A METHODOLOGY TO ASSESS THE AIR FORCE MATERIEL COMMAND'S EXERCISE SUPPORT PROGRAM'S WEIGHTING SCHEME

THESIS

Gilbert D. Ness, 2nd Lieutenant, USAF

AFIT/GOR/ENS/96M-10

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

DTIC QUALITY INSPECTED 4

Wright-Patterson Air Force Base, Ohio

19970502 244

AFIT/GOR/ENS/96M-10

A METHODOLOGY TO ASSESS THE
AIR FORCE MATERIEL COMMAND'S
EXERCISE SUPPORT PROGRAM'S
WEIGHTING SCHEME

THESIS

Gilbert D. Ness, 2nd Lieutenant, USAF

AFIT/GOR/ENS/96M-10

The views expressed in this thesis are those of the author and do not reflect the official

policy or position of the Department of Defense or the U.S. Government.

AFIT/GOR/ENS/96M-10

A METHODOLOGY TO ASSESS THE AIR FORCE MATERIEL COMMAND'S

EXERCISE SUPPORT PROGRAM'S WEIGHTING SCHEME

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

Gilbert D. Ness, B.S.

2nd Lieutenant, USAF

March 1996

THESIS APPROVAL

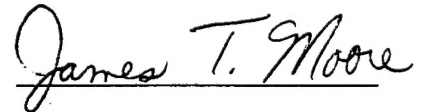STUDENT: Gilbert D. Ness, 2nd Lieutenant, USAF   CLASS: GOR-96M

Thesis Title: A METHODOLOGY TO ASSESS THE AIR FORCE MATERIEL
COMMAND'S EXERCISE SUPPORT PROGRAM'S
WEIGHTING SCHEME
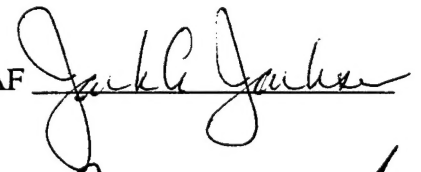
Defense Date: 4 March 1996

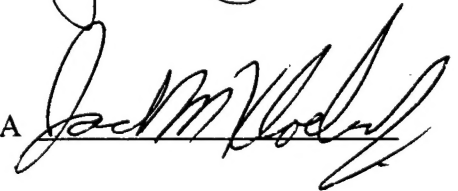| Committee:	Name/Title/Department	Signature
Advisor	JAMES T. MOORE, Lieutenant Colonel, USAF Associate Professor of Operations Research Department of Operational Sciences Graduate School of Engineering	
Reader	JACK A. JACKSON Jr., Lieutenant Colonel, USAF Assistant Professor of Operations Research Department of Operational Sciences Graduate School of Engineering	
Reader	JACK M. KLOEBER Jr., Lieutenant Colonel, USA Assistant Professor of Operations Research Department of Operational Sciences Graduate School of Engineering	

Acknowledgments

I would love to give special thanks to the large number of people that have helped me come this far in my journey through life. However, this page is too short to recognize all of those that have had such a great influence on me, so I must limit myself to those who have helped me through this one period in time. Most important of all, I thank my mother, father, and grandmother for giving me the encouragement and support to make it through this graduate program.

I must especially thank my thesis committee for their guidance throughout this entire ordeal. A great magnitude of thanks goes to LtCol Moore, who through his advisory role, took my hand and lead me down a very dark and windy path. Many thanks also to LTC Kloeber and LtCol Jackson for the insight to clear the path during troubled times. I cannot leave out TSgt Scullion who was always available at a moments notice to help understand the AFMC problem. The remaining thanks go out to all of my friends here that have struggled, partied, passed, failed, brewed, trumped, rucked, mauled, and graduated with me here in Dayton, Ohio.

<div align="right">Gilbert D. Ness</div>

Table of Contents

List of Figures

List of Tables

AFIT/GOR/ENS/96M-10

Abstract

This research performs analysis on the Air Force Materiel Command's Exercise
Support Program's Weighting Scheme so that decision makers can create a more efficient
and effective Resource Allocation - Integrated Process Team (RA-IPT) manpower
allocation process. A linear programming (LP) model was derived from the Exercise
Support Program (ESP) to assign manpower reductions throughout the Mission Element
Board (MEB) organizational level in AFMC. Parametric analysis was performed on the
LP model by simultaneously changing multiple objective function coefficients with the use
of various direction vectors. These direction vectors were applied with the use of an
incrementally increasing scalar step size and initiated from two initial states of objective
function coefficient values. Throughout the analysis, multiple bounds on the LP model's
constraints and different scalar step sizes were applied. Results of the analysis indicated
that there were specific situations in which changes in the relative weighting scheme did
effect manpower allocations to the MEBs. These results also indicated that this analysis,
along with the ESP model, could allow the decision makers in the RA-IPT to become
more efficient in their manpower allocation process.

A Methodology to Assess the Air Force Materiel Command's Exercise Support Program's Weighting Scheme

I. Introduction

1.1 Background

As Air Force resources decline, decision makers need tools to help prioritize programs and make resource-allocation decisions in terms of both dollars and manpower. These decisions must be made in an effort to meet the funding constraints while still accomplishing the missions and objectives of the Air Force. Such a decision requires careful consideration of an integrated framework of economic, political, and war fighting factors.

At Headquarters Air Force Materiel Command (HQ AFMC), the manpower reduction process involves generating reduction alternatives through a Resource Allocation - Integrated Process Team (RA-IPT). This process has a hierarchical nature where the RA-IPT receives input from five specific Mission Element Board (MEB) members who, in turn, receive input from the lower levels of their organizational structure. The RA-IPT then takes each MEB input and produces reduction alternatives. Given these alternatives, HQ AFMC/XPM (Manpower Division) generates numerical solutions, assesses the impacts, and ensures all constraints are met. The feasible alternatives are then directed back to the RA-IPT for further review. This process continues until one viable solution is found and agreed upon by all members of the RA-IPT.

The current process is limited in its effectiveness mainly because it is very time consuming for the five distinct MEBs to reach agreement on any one reduction alternative. Any one of the five MEBs may seem to be favored over another in an alternative's allocations, causing turmoil within the process and generally leading to the fallback solution of a straight percentage cut over all the organizations (the Peanut-Butter Spread Method). This straight cut method leaves much to be desired in terms of generating an actual cost- effective reduction since high priority programs are cut the same percentage as low priority programs [2]. To make this a better process, AFMC decision makers require two important decision making tools: a decision support system to aid them in allocating resources and making manpower decisions given budget constraints, and a way in which to evaluate those resulting allocations.

Previous thesis work in this area was done in 1995 by Captain Steve Bishop. Bishop created the Exercise Support Program (ESP) decision support system for decision makers, such as the RA-IPT leaders, to use. ESP allows the user to define different weighting schemes for the MEBs and their respective functional and organizational subcommands. These weighting schemes are then used within ESP's linear program to drive the generation of possible allocation alternatives [1:14].

The sensitivity analysis performed by Bishop on individual weights, which can represent the different elements of the entire RA-IPT process, represented the effects of changing one weight while the others remained constant [1:44]. However, the scope and focus of his research did not include the analysis of simultaneous changes in multiple objective function coefficient weights. This type of post-optimality analysis is referred to

as parametric analysis. Such analysis allows the user to determine the "trade-offs" between the weights by comparing the changes in the optimal solution as simultaneous changes in the weight parameters are made. By doing so, the decision maker can develop a better understanding of the model and the effects of the inputs. Combining the ESP decision support system and results from parametric analysis, the RA-IPT process could become a more effective and efficient process.

1.2 Problem Statement

AFMC decision makers currently have access to a complex decision support system (ESP) devoted to aiding them in allocating manpower during an RA-IPT process. However, changing attitudes within the RA-IPT process can lead to changing input parameters in ESP and changing overall manpower allocation strategies. The decision makers in the RA-IPT would like to know how dependent the ESP output is on these changing attitudes. Therefore, they require analysis encompassing the effects of their input and changes in their input to the ESP model.

1.3 Research Objective

The goal of this research is to allow decision makers to better understand the effects of simultaneously changing the input parameters of ESP. Specifically, the parameters of interest include the weights associated with each MEB with respect to the weights given to other MEBs. A secondary objective of the research is to create a model that a decision

maker, in the RA-IPT process, can use to produce the optimal manpower allocations at the MEB level given a set of user-defined weights.

1.4 Scope

This research does not try to generate complete AFMC manpower allocation alternatives nor does it try to evaluate the alternatives generated since previous research has already done so. It is solely intended to help AFMC decision makers understand the effects of his or her inputs into the ESP manpower allocation generator. The parametric analysis performed will indicate the relationships between multiple changes in the weighting scheme for each MEB and the different allocation alternatives. In doing so, the decision maker will hopefully be able to attain better resource allocation alternatives by knowing, beforehand, the effects of his or her input.

Through the ESP Program, the decision makers can quantify their opinions on the importance of each of the different MEBs and obtain the resulting allocation alternative. We would also like the decision makers to be able to know the effects of altering the importance of the MEBs before they make their final decision. For the AFMC problem, such analysis could prove useful when the MEB decision makers disagree in the RA-IPT. If the analysis were to determine that small simultaneous changes in the weighting scheme drove large changes in the alternatives, then decision makers would carefully need to assess their MEB ratings. However, if the analysis indicated that simultaneous changes in the weighting scheme had little effect on the alternatives, then little debate might be called for and time could be saved in the process.

4

The Microsoft EXCEL spreadsheet included will allow the decision maker to forego running the highly detailed ESP program and see a general result of the weighting scheme he or she chooses. ESP can then be used later to determine the detailed manpower allocations to be made given a number of reductions throughout AFMC.

1.5 Overview

Chapter II reviews past research and begins to develop the problem at hand. It begins by describing the current RA-IPT process used by AFMC to allocate manpower reductions. Bishop's ESP decision support system is also reviewed. Next, a short review of the literature pertaining to parametric programming/analysis follows. Finally, the software and methodology linked to parametric programming is described.

Chapter III describes the methodology of the parametric analysis. First, the actual model is derived from the ESP linear program. Then, the data is outlined and the method of analysis is illustrated. Finally, the initial states of objective function coefficient values are determined and the direction vectors to be studied are derived.

Chapter IV presents the analysis accomplished and discusses the resulting allocation effects.

Chapter V presents the final conclusions and gives recommendations for further study in this area.

Appendix A contains the FORTRAN code written to perform the parametric analysis in CPLEX, while Appendix B presents the numerical results of the analysis.

II. Review of Previous Research

2.1 Introduction

The manpower allocation problem at Headquarters Air Force Materiel Command (HQ AFMC) has called for the development of different manpower models and analyses in hopes of shedding some light on the best way in which to spread the recent manpower reductions throughout the command's five MEBs and their organizations. This chapter discusses the current allocation process and one of the manpower models created to assist that process. It then describes a method to evaluate the effects of that model given simultaneous parameter changes in the input.

2.2 The Resource Allocation - Integrated Process Team

Within AFMC, there exist five main functional areas into which every task performed by AFMC personnel can be categorized. These five functional areas are defined as Mission Element Boards (MEBs) and include: Base Operating Support (BOS), Science and Technology (S&T), Support and Industrial Operations (S&IO), Test and Evaluation (T&E), and Program Management (PM). These MEBs represent the top tier of the functional and organizational hierarchy within AFMC. Each MEB is headed by the specific directorate of AFMC which is most closely associated with the MEB's function. For example, the BOS MEB is headed by the Civil Engineering (CE) directorate while the PM MEB is headed by the Directorate of Requirements (DR).

The next tier would include the centers within each MEB. This 'center' level is mainly composed of three test centers, four product centers, three laboratories, and five air logistic centers. Other centers exist, but they are not included in the research due to their limited size and responsibilities. Each center's primary mission can be categorized into at least one of the MEBs, with the general operations for all of them falling into the BOS MEB. While the MEBs do not actually own the centers and do not have direct command authority over them, each MEB does own the resources in the centers that are associated with that respective MEB's primary mission.

The relationship between the centers and the MEBs can be seen in the following 'stovepipe' diagram depicted in Figure 2.1. The centers associated with each MEB are seen above the MEB's column while universal functions within AFMC are seen horizontally through the MEB columns. The horizontal lines cutting through the MEBs indicate that the functions are not exclusive to any one MEB. AFMC focal points indicate the two-letter organization that is established as the ultimate head of the MEB. The hierarchy can be broken down even further by two- and three-letter command structures within the centers. However, due to the scope of this research, only the top tier of the command structure is analyzed in the application of parametric analysis.
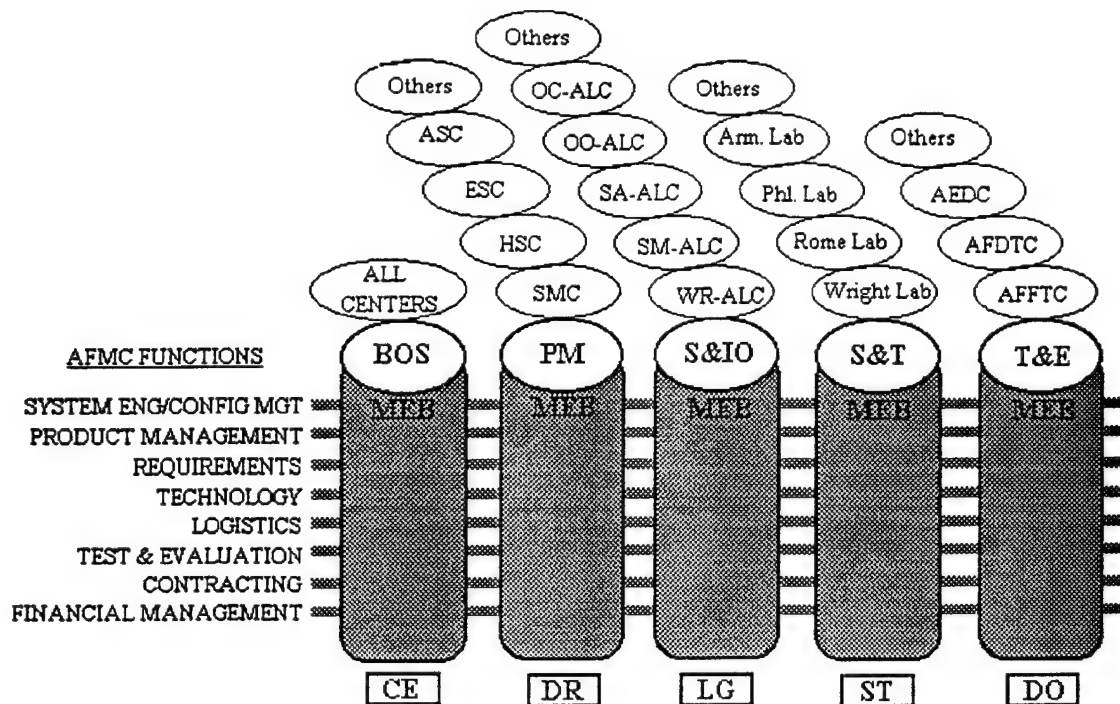
Figure 2.1. MEB Manpower Model Command Management Framework

The current AFMC manpower allocation process begins with a reduction established
by the Air Staff and administered directly by HQ AFMC/XPM, the Manpower Division at
HQ AFMC. This reduction most often spans a five year horizon and includes both
specific cuts (programmatic) and general cuts (non-programmatic). Since programmatic
cuts are clearly set to effect a specific element of AFMC and are not a part of the RA-IPT
process, they were not taken into account by Bishop, and they are not incorporated in this
research. AFMC then passes the non-programmatic reductions to the RA-IPT, composed
of five upper level decision makers, each representing one of the MEBs. The RA-IPT's
role is to decide how the manpower cuts will be allocated among the MEBs. They are

8

also tasked with determining how the cuts will be spread among their own competing

functions and organizations at the lower levels of the MEB; therefore, they solicit

feedback from within their own MEB's structure throughout the process. A general

depiction of this process can be seen in Figure 2.2.

CURRENT RA-IPT PROCESS
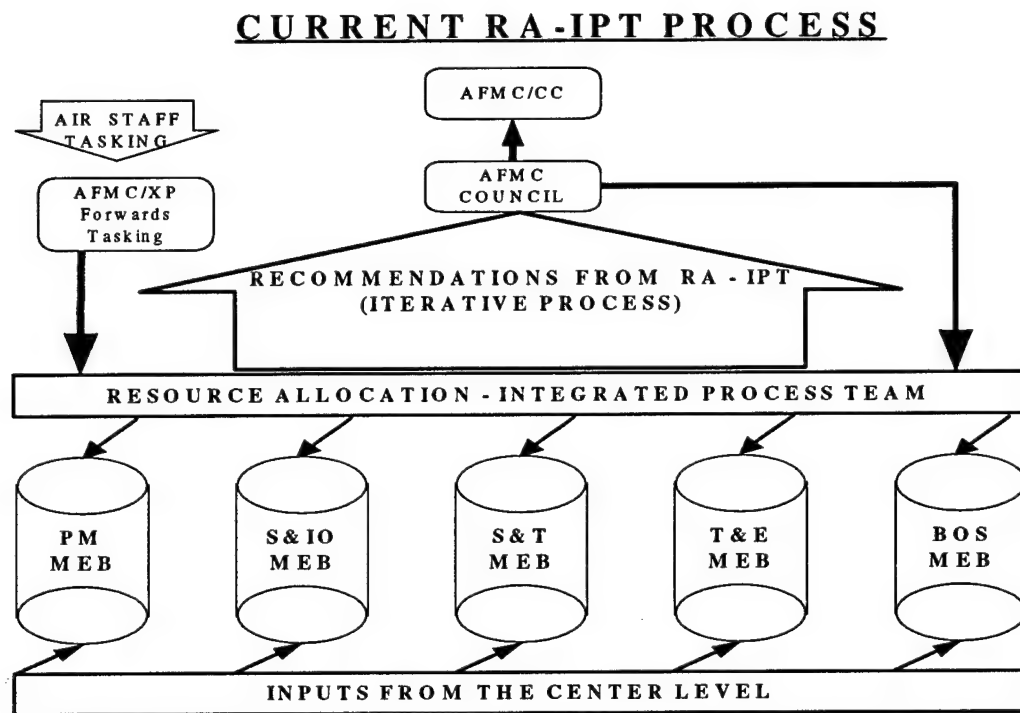
Figure 2.2. Current RA-IPT Process Diagram

The process revolves around the MEB decision makers and their ability to

cooperatively make the most efficient manpower allocations. Efficiency is defined as

cutting manpower from organizations whose effectiveness is impacted the least by these

manpower reductions. However, as the process is channeled between the MEBs and the

lower levels of the structure, larger numbers of functions and organizations are allowed into this 'battle arena' in which compromise is the only way to reach a solution.

Normally, this decision making process becomes both difficult and time consuming when these competing organizations, at all levels, refuse to agree or indorse manpower allocations in which their organization's manpower is cut disproportionately compared to another organization's manpower. As a consequence of this inability to reach a solution, often the final solution is to take an equal proportion of manpower away from each organization in their respective level of the command structure. This is known as the "Peanut-Butter Spread" method which is not an efficient approach to manpower allocation [12].

2.3 Previous Efforts in Manpower Allocation

Previous efforts have devoted much time to the aspect of applying linear programming- type methods to manpower allocation problems. Karthikeyan and Krishnaswamy applied linear programming involving proportionality constraints to a manufacturing organization with multiple assembly and subassembly requirements [10]. Gass describes the Military's use of: Markov models to forecast personnel inventory levels, network flow formulation to assign personnel, and network-like goal programming models to plan future allocations [8].

In a recent Interfaces article on the Army's approach to manpower reductions, Eiger, Jacobs, Chung, and Selsor unfold the U.S. Army's manpower decision support system as a combined linear optimization-simulation flow model that has the ability to

10

reflect in greater detail previous linear models [5]. The goal of the Army in this problem is similar to that in the RA-IPT process, but the Army's problem is a bit more complicated. Their goal is to manage their personnel (enlisted only in this problem) in a way that minimizes shortages or overages while still meeting strength, skill, and grade requirements [5:59]. The Military Occupational Specialty Level System (MOSLS) is the decision support system used to not only solve the problem but also to answer 'what-if' questions [5:57]. This system uses a network flow problem to model the personnel assignments and requirements. The network flow model consists of optimizations based on separable linear goal programming [5:64]. However, since the problem is too large to formulate as a pure network, it is decomposed into more general network flows which causes hidden problems in the interpretation of the results. Therefore, the optimization was augmented by a simulation that could adapt itself to more accurately reflect policies in personnel management [5:67].

2.4 The Exercise Support Program [Bishop, 95]

In an effort to help the top level (MEB) decision makers generate better alternatives for manpower allocations, Bishop created the Exercise Support Program (ESP). The ESP decision support system was created in Microsoft EXCEL[1] using the Visual Basic language and taking advantage of the Microsoft EXCEL Solver applications [6]. ESP requires the decision makers' quantification of the importance of each of the MEBs with respect to each other, the centers with respect to each other, and the two-letter

---

[1] The use of Microsoft EXCEL software was a user requirement set by HQ AFMC/XPM.

organizations with respect to each other. Once these opinions or 'weightings' are obtained, ESP then places them into a linear program designed to generate manpower allocation strategies based on these weightings and current manpower allocations.

ESP queries data from the Command Manpower Data System (CMDS). This system, continuously updated by HQ AFMC/XPM, contains the manpower allocation currently assigned AFMC for the current year and five years into the future. The data is broken down by the MEB, center level, and two-letter level organizational structure. Within those elements, the data is then divided into the categories of grades: officer, enlisted, civilian, and contract manpower equivalents (CMEs). Under the current process, the Air Force does not cut manpower from the CMEs directly through the RA-IPT process; therefore, Bishop did not consider the CME data.

Bishop chose to form a linear programming (LP) problem with manpower allocation strategies being optimized. Decision variables included the manpower allocation, by grade, for each two-letter organization within each center within each MEB. Due to the large number of variables involved and the limitations of the Microsoft EXCEL software, Bishop had to decompose the problem into 150 subproblems that could be solved by the Microsoft EXCEL optimizer.

The decomposition took place at the different levels of decision making within the RA-IPT. The first optimization allocated manpower among the MEBs. These allocations were then used as input parameters in the second optimization at the center level. These allocations, in turn, were used as input parameters in the final optimization among the two-letter organizations. By breaking down the original linear programming problem into

12

these subproblems, an optimal solution is not guaranteed [1:15]. The scope of this current research requires that only the first optimization at the MEB level be analyzed. The following linear program and variable definitions are from Bishop's work [1:16]:

m = Number of MEBs (5)

c = Number of centers subordinate to the MEBs (dependent upon specific MEB)

t = Number of two-letter commands subordinate to the centers (dependent on specific center)

r = Number of grades (Officer, Enlisted, or Civilian)

y = Number of years in the planning horizon (5)

X_{mry} = Number of manpower authorizations in each MEB m by grade r for each year y **after** the reductions

Z_{mry} = Parameter designated for the authorized manpower in each MEB m by grade r for each year y **before** the reductions

w_{my} = Nonnegative parameter designated for the weight assigned to each MEB m by year y

N_y = Number to be reduced from the current authorized manpower by year y

L_{ry} = Parameter designated for the percentage of authorized manpower **allowed** to be reduced by grade r for each year y

R_y = Parameter designated for the officer-to-enlisted ratio in year y

$$\mathbf{MAX} \quad \text{STRENGTH} = \sum_m (W_{my} \sum_r X_{mry}) \qquad \text{for each } y \qquad (1)$$

SUBJECT TO:

$$\sum_r \sum_m X_{mry} = \sum_r \sum_m (Z)_{mry} - N_y \qquad \text{for each } y \qquad (2)$$

$$(1 - L_{ry})(Z)_{mry} \leq X_{mry} \leq (Z)_{mry} \qquad \text{for each } m,r,y \qquad (3)$$

$$\sum_r (X_{(1)ry} + X_{(2)ry} + X_{(3)ry} + X_{(4)ry} + X_{(5)ry}) =$$

$$\sum_m (X_{m(civ)y} + X_{m(enl)y} + X_{m(off)y}) \qquad \text{for all } y \qquad (4)$$

$$R_y \sum_m X_{m(off)y} \leq \sum_m X_{m(enl)y} \qquad \text{for all } y \qquad (5)$$

$$X_{mry} \geq 0 \qquad \text{for all } m,r,y \qquad (6)$$

The objective function (1) maximizes the weighted sum of the manpower allocations to be made to each MEB, by each grade, for each year. The weighting parameters, which only weight the MEBs by the respective year, are coded in ESP as input from the decision maker. However, if the decision maker is unwilling to assign one or multiple weights, then ESP assigns the weights based on the level of manpower currently assigned to the MEBs for that respective year [1:17]. No restrictions are placed on the weights that can be assigned the MEBs, except that negative values would not make sense. The higher the weight, the more 'important' the MEB with respect to the other MEBs, given that the others have lower weights assigned.

The first constraint (2) ensures that the manpower reduction for each year (N_y) is met by forcing the sum of the manpower allocations to be assigned over each grade within each MEB over each year to equal the sum of those current allocations minus the

appropriate level of cuts. The second constraint (3) mandates that the cut be an actual cut and not an addition by bounding the upper limit on the allocations to be made with the current manpower levels. This constraint also sets a lower limit on the allocations to be made by not letting them fall below a percentage level set by the decision maker. The third constraint is a "sanity" check. It ensures that the sum of allocations to be made for each MEB each year, by each grade, equals the sum of allocations to be made for each grade each year, by each MEB. The fourth constraint (5) ensures that the officer-to-enlisted ratio remains intact throughout the process. The final constraint (6) is a nonnegativity restriction for the allocations.

Smith complemented Bishop's work by creating a method to evaluate the manpower allocations generated [12]. Her work was based on results gathered from surveying the entire two-letter level of AFMC. These results led her to create an additive value function derived from utility functions created for each two-letter organization. The final model was a predictor of how well AFMC could effectively accomplish its mission after specific manpower reductions were made.

2.5 Parametric Analysis

As the scope of this current research states, the decision makers need to understand the effects of altering their opinions on the importance of the different MEBs. The type of analysis which can determine the effects of changing attitudes within the RA-IPT allocation process is referred to as parametric analysis. Sensitivity analysis studies the effects of varying one parameter input at a time, but parametric analysis studies the effects

of varying multiple input parameters simultaneously. This allows a multitude of

applications. One of the main goals of parametric analysis is to interpret the appropriate

trade-offs between multiple input parameters [9:98]. For the AFMC problem, these

potential trade-offs could describe underlying relationships between the MEBs. Once

these relationships are understood, MEB chiefs could facilitate the RA-IPT process by

knowing critical relationships between MEBs.

Parametric analysis uses parametric programming, which begins with the original

linear program and restructures the objective function. Generally, the linear program

would take the form:

$$\text{Maximize } Z = \mathbf{cx}$$

$$\text{Subject to } \mathbf{Ax} \leq \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}$$

where $\mathbf{c}$ is a (1 x n) row vector of objective coefficients, $\mathbf{x}$ is an (n x 1) column vector of

decision variables, $\mathbf{A}$ is an (m x n) matrix of constraint coefficients, $\mathbf{b}$ is an (m x 1)

column vector of the right-hand-side constraint values, and $\mathbf{0}$ is an (n x 1) column vector

of zeros. When changes are made to objective function coefficients using parametric

programming, the constraints remain the same but the Greek letters λ and θ, representing

the parameters of parametric analysis, are added to the objective function in the form of:

$$\text{Maximize } Z = (\mathbf{c} + \lambda\theta)\mathbf{x}$$

where θ is a (1 x n) row vector representing the direction for changing the objective

function coefficients and λ represents the scalar variable which we increase by

incremental steps [9:308]. The objective of parametric programming is to then derive the optimal solution of the newly structured linear program as a function of λ, forming a piecewise linear, convex solution set [9:308].[2]

In order to apply parametric programming to the AFMC problem, and for the sake of simplicity, we must first make two assumptions [3:136]. The first is that λ is greater than or equal to zero. If necessary, the negative aspects of λ can be realized by changing the θ vector. For this research, no negative values will be used since negative weightings are not desirable. The second assumption is that the original problem Bishop solved is not degenerate and has a finite optimal solution.

Determining the objective function's value as a function of λ requires varying the value of λ incrementally from zero to a set upper bound. This will give the effects of *magnitudinal* changes between the weights associated with each of the MEBs. To analyze the *relationships* between the MEB weights, we must run these experiments while altering the vector θ. The level of effort required for this type of analysis dictates the use of optimization software such as CPLEX. CPLEX gives the ability to create the programs in FORTRAN code; thereby allowing consecutive optimizations to be based on previous results. This becomes a necessity for the manpower allocation process, given a time horizon spanning five years, since upper bounds must be changed to reflect cuts made in the previous years' allocations.

---

[2] At $\lambda = 0$, the transformed objective function is the same as the original objective function, and the optimal solution remains the same as in the original linear program.

III. Methodology

3.1 Introduction

This chapter contains the methodology by which the AFMC problem is analyzed. Modifications made to Captain Bishop's model are examined and the CPLEX model is presented. The AFMC manpower data is introduced and is followed by the description of the analysis to be performed. The direction vectors to be applied in the analysis are then developed.

3.2 Modifications to the ESP Linear Programming Problem

It was determined that the ESP Model was still acceptable in terms of its linear programming capabilities and that integer programming was not a necessity for this type of analysis: rounding manpower allocations was sufficiently accurate [11]. Other manpower allocation models at AFMC use this same approach. However, before initiating the parametric analysis, other aspects of Bishop's original linear program needed modification in order to meet the objectives of this research.

The scope of this research dictates that only the MEB level of the AFMC hierarchy be studied. This, and the fact that MEB decision makers are primarily concerned with their overall manpower total rather than the individual totals of their officer, enlisted, and civilian populations [11], means that the decision variables only need to represent the entire manpower allocation for the respective MEB each year. This reduces the number of decision variables from 75 to 25. (Originally, there was a decision variable for each of the

grades within all five MEBs for each of the five years; now the decision variables represent each of the five MEBs in each year.) This also allows the elimination of the sanity check constraint since individual grade totals are no longer needed for each MEB. Supporting this point is the fact that although the officer-to-enlisted ratio is an overall goal within the military, it is not a set standard when applying AFMC manpower allocations in the RA-IPT process at the MEB level [7]. Therefore, for this research, the officer-to-enlisted ratio constraint was also removed from the ESP model.

3.3 The CPLEX Linear Programming Model

The resulting model (after the modifications) then becomes, as seen below, a bounded knapsack problem optimized for each year. This linear problem must be solved for each year, and in correct order, to achieve the proper and continuing relationships among the consecutive years. A single iteration in the FORTRAN driven model consists of these five consecutive optimizations with a common set of initial objective function coefficients for each. (An example of the FORTRAN code can be seen in Appendix A.)

m = Number of the mission element board member (BOS = 1, PM = 2, S&IO = 3, S&T = 4, T&E = 5).

y = Year in the planning horizon (1-5).

X_{my} = Number of manpower authorizations in MEB m for each year y after the reductions.

$\mathbf{Z_{my}}$ = Parameter designated for the authorized manpower in MEB m for each

year y before the reductions.

$\mathbf{w_{my}}$ = Nonnegative parameter designated for the weight assigned to MEB m

by year y.

$\mathbf{N_y}$ = Total number to be reduced in AFMC from the current authorized manpower by

year y.

$\mathbf{L_y}$ = Parameter designated for the percentage of total authorized manpower in AFMC

allowed to be reduced each year y.

λ = Scalar representing the amount by which the rate of change will occur for each

incremental step in the analysis process.

Θ_m = Parameter designated for the relative rate at which the objective function

coefficient for MEB m changes with respect to the other coefficients. Therefore,

the set of Θ represents the direction vector for the parametric analysis.

$$\mathbf{MAX} \quad \mathbf{STRENGTH} = \sum_m (w_{my} + \lambda^* \Theta_m) X_{my} \quad \text{solved for each } y = 1...5 \quad (1)$$

SUBJECT TO:

$$\sum_m X_{my} = (\sum_m Z_{my}) - N_y \quad (2)$$

$$(1 - L_y) Z_{my} \le X_{my} \le Z_{my}, \text{ for each } m \quad (3)$$

The objective function maximizes the "strength" of the manpower allocation based
on the weighting coefficients chosen by the decision makers. This objective function value
has no true meaning to the decision maker. Its only ramification is to drive the manpower

allocations to the best levels given the decision maker's beliefs on the relative importance of each of the MEBs. Once a customized vector of the objective function coefficient values is obtained from the decision maker, the CPLEX model solves one iteration of the process and writes the optimal manpower allocation to a data file. The model then proceeds to the next iteration where the objective function coefficient values are increased or decreased according to the direction vector and scalar step size. This process continues until the set number of iterations are performed and the desired range of λ is examined.

The single constraint forces the mandated cuts to be made each year. This is accomplished in the FORTRAN code by taking the current manpower levels associated with each year and subtracting the fixed levels of the reductions (N_y). Within each iteration, each optimization takes into account the manpower reductions made in the previous optimizations.

Initially, the manpower authorizations for each MEB are bounded above by the current manpower authorizations made in some previous RA-IPT process, but they are also subject to any cuts made in the current iteration's process. For example, the upper bounds for the BOS MEB in year 5 must reflect cuts assigned in the years 1 through 4. To facilitate this operation, calculations of the total reductions made for each MEB in each year are stored within each iteration and are subtracted from the current manpower authorization for the respective MEB in the particular year. This ensures that the upper

bound for an MEB manpower allocation can never be larger than what is currently

assigned minus the reductions made in the previous year's optimization.[3]

The manpower allocations are then bounded below by a set percentage (given by the

decision maker) of the upper bound. This ensures that no one MEB is cut below some

tolerance level. However, the decision maker must set this percentage relative to the

manpower reductions being made. If this is not done, then the possibility of an infeasible

solution exists. An example would be the mandatory reductions not being met because the

percentage is set too high. This constraint also ensures that no one MEB takes all the cuts

based solely on the fact that it might have the lowest weighting in the objective function.

These lower bounds also establish the nonnegativity conditions for the manpower

authorizations, thereby eliminating the need to include them in the linear program.

3.4 Manpower Data

The AFMC data used in this research comes from the CMDS. This database

contains the approximately 160,000 manpower authorizations currently in AFMC,

including the CME population. Focusing on the most current data available, September

1995, while excluding the CMEs, HQ AFMC personnel, and miscellaneous unit personnel,

the total current AFMC manpower authorizations are approximately 115,000. Using

software designed as a database manager, the data was categorized by MEB for the

---

[3] There are two separate instances in the current data that include a higher manpower authorization for a
consecutive year's allocation. This fact, however, does not change the capability of the upper bound
constraint, it merely allows two instances where, if no cuts are determined for that MEB in the previous
year's optimization, an increase in manpower is possible for that MEB in the following year.

purpose of this research. In compliance with the five year planning horizon set in the RA-IPT process, the total manpower authorizations, by MEB, were obtained for the current year and four years into the future, as seen in Table 3.1. (The previously mentioned case of an MEB actually gaining manpower authorizations can be seen between year 1 and year 2 and between year 4 and year 5 for the S&T MEB.)

Table 3.1 Current Manpower Authorizations By MEB

MEB	Year 1	Year 2	Year 3	Year 4	Year 5
BOS	35438	35200	34759	34601	34403
PM	31864	30531	30136	29817	29563
S&IO	32787	31032	28657	28491	28231
S&T	6895	6908	6697	6688	6689
T& E	8643	8390	8038	7964	7898
Totals	115627	112061	108287	107561	106784

3.5 Analysis Methodology

The majority of parametric analysis done in previous research focuses on the optimal *value* of the objective function as simultaneous parameter changes are performed in the problem. Since the scope of this research does not attribute a meaning to this value, the analysis performed concentrates on the optimal values of the decision variables as simultaneous changes are made within the problem. The decision variables represent the manpower allocations to be made to each MEB.

The analysis begins by determining the initial objective function coefficient vectors to be analyzed. Each vector includes the objective function coefficient values for the entire five year planning horizon and represents an initial condition that might be expected

23

as input by decision makers in the RA-IPT process. Once the initial objective function coefficient vectors are realized, the next step is to create the direction vectors and step sizes associated with those direction vectors. The direction vectors are representative of the possible changes made by decision makers during the RA-IPT process. These direction vectors may either be continuous over the five year planning horizon or specific to each year within the horizon, depending on the analysis to be made. The application of a direction vector to an initial objective function coefficient vector constitutes one run of the parametric analysis.

Once these initial vectors are determined, the FORTRAN program can be modified to model the impact of each direction vector on each initial objective function coefficient vector. A single run of the parametric analysis begins by solving the five linear optimizations (representing each year in the planning horizon) with an initial set of objective function coefficient vectors (each year's will differ slightly). This is considered one iteration. The next iteration then multiplies the direction vector by the scalar step size, adds this to the set of initial objective function coefficient vectors, and solves the five linear optimizations. This process continues until the range of the scalar step size, λ, is covered, thereby ending the run. For this analysis, the desired range of λ will be covered in 100 iterations.

Output from each of the runs performed consists of the optimal objective function values and the optimal decision variable values for each year in each iteration. Microsoft EXCEL software is then used to create pivot tables and graphs of the optimal manpower allocations for each MEB every year over the range of λ. This creates five graphs for

each possible combination of initial objective function coefficient vectors and direction vectors. The graphs indicate changes in the manpower allocations as the step size for the direction vector increases. These changes in the manpower allocations are deemed points of interest. Each of these points represents a relative weighting scheme in the objective function that might have meaning to the decision makers in the RA-IPT process.

3.6 Development of Direction Vectors

The direction vectors chosen for the parametric analysis had to represent possible changes in attitudes by the RA-IPT decision makers on the relationships between the MEBs. Since a multitude of events could lead to various changes in attitude, as many plausible scenarios as possible were analyzed. These direction vectors can be seen in Table 3.2. For the initial analysis, a single direction vector is carried through all five years with no changes in the vector between the years.

Table 3.2 Directional Vectors Analyzed

Ref #	Vector Description:	BOS	PM	S&IO	S&T	T&E
1	All MEBs Equal (Test case)	1	1	1	1	1
6	Linearly Ordered	5	1	4	3	2
11	Weighted by Utility of Manpower Loss	0.9	1.5	0.4	4.1	3.0
	BOS Not Important					
2	All MEBs Equal	0	1	1	1	1
3	Important Acquisition Period	0	1	0	1	1
4	Important Acquisition Period With Strong PM	0	2	0	1	1
5	Important System Maintenance Period	0	0	1	0	0
	BOS Extremely Important					
7	All MEBs Equal	10	1	1	1	1
8	Important Acquisition Period	10	1	0	1	1
9	Important Acquisition Period With Strong PM	10	2	0	1	1
0	Important System Maintenance Period	10	0	1	0	0

Direction vector (1), consisting of all ones, was used to verify that the parametric program was functioning properly. This direction vector adds the same amount of change to each of the MEBs during each increment of λ. Since the vectors are normalized during each of the optimizations, this direction vector will cause the objective function coefficients to converge to a point where all of the MEBs are weighted equally. This then allows the MEBs that are initially less important to become more important, while the MEBs that were initially more important become less important.

The next direction vector (6) was chosen to represent a possible current ranking of the MEBs by the RA-IPT decision makers. The five MEBs were given a value of one to five, with five being the most important, based on the views and opinions of AFMC personnel.[4] This vector would allow the more important MEBs' objective function coefficient values to grow faster than the less important MEBs' objective function coefficient values. Eventually, as the iterations increased, it would be expected that the most important MEBs (S&T and T&E in this case) would take the fewest reductions.

The next direction vector (11) was derived from the results of Smith's work. Her utility functions were aggregated at the two-letter level by MEB. Each two-letter organization was given a weight based on its current manpower strength within the respective MEB. Calculations were then made using Smith's additive value function to determine the effects on total utility of a set number of reductions (100) for each MEB [12:29]:

---

[4] These rankings **do not** reflect the opinions of Air Force Materiel Command and its leadership. They represent notional data gathered from various members of AFMC during personal interviews and do not reflect the official policy or position of AFMC.

$$U(x_a) = \sum_{m=1}^{5} W_m \cdot \sum_{c=1}^{C_m} W_{mc} \cdot \sum_{t=1}^{T_c} W_{mct} \cdot U_{mct}(x_{amct})$$

(1)

where:

a = manpower allocation alternative

w = weighting factors

m = mission element area, $1 \le m \le 5$

c = center, C_m = number of centers with manpower resources in mission element area m

t = two-letter organization, T_c = number of two-letter organizations in center c

$U_{mct}(x_{amct})$ = mission effectiveness value of two-letter organization given the manpower resources allotted by alternative a

The differences in utility were then normalized in order to compare the relative losses of utility for each MEB, as presented in Table 3.3. Assuming that this utility measures the 'effectiveness' of each MEB [12:24], it would then be beneficial to keep those MEBs with expected high utility losses free from further reductions. Therefore, a direction vector (11) was created representing the relative losses of utility based on the normalized differences. As λ increases, S&T and T&E would be expected to take fewer reductions while BOS, PM, and S&IO would take more. (The normalized differences were multiplied by a factor of ten to keep this direction vector at the same order of magnitude as the other direction vectors.)

Table 3.3 Utility Values and Differences by MEB

Manpower	BOS	PM	S&IO	S&T	T&E
Current (1994) Utility	90.140	91.731	75.568	90.945	88.280
Utility of MEB After Loss of 100	89.844	91.255	75.437	89.659	87.334
Difference:	0.296	0.476	0.131	1.286	0.946
Normalized Difference:	0.094	0.151	0.042	0.410	0.302

The remaining eight direction vectors were divided into two groups: those that considered BOS as an extremely important MEB and those in which BOS was not as important. This division was in response to the notion that the BOS MEB either could or could not afford to take additional manpower reductions in the future due to reductions made in the past [7]. To represent the situation where the BOS MEB could not afford reductions, the direction vector assigned a value of 10 to BOS, which is approximately an order of magnitude greater than the other values in the direction vectors. With this relatively large rate of change, the BOS MEB would not be expected to take any manpower reductions, unless constraints forced them. However, a contrasting view would note that BOS is the largest of the MEBs and may be able to take a greater number of the reductions. To reflect this in the direction vector, the value of 0 was given to BOS for this group, indicating that the BOS coefficients would never increase in value and would therefore be expected to take as many of the reductions as possible as the scalar step size increased. These contrasting vectors, where the BOS MEB was given the value of 10 or 0, are to verify the results of the parametric program and provide a set of

interesting cases. The remaining MEBs have duplicate values in the respective direction vectors of the two different groups.

The first direction vector of each group (2 and 7) consists of the PM, S&IO, S&T, and T&E MEBs having the same relative change in the objective function coefficients while the BOS either remained at its initial level, or grew faster than the rest. These vectors were created to represent the situation where the RA-IPT considers the impacts of the relationship between the BOS MEB and the others while keeping the remaining MEBs constant in their relationships. If the BOS MEB had a direction value of 0 in the vector, then as the value of λ increased, it would be expected to take more and more of the reductions until the maximum allowed by the bounds of the problem are reached. On the other hand, if the BOS direction value was 10, then it would be expected to take fewer cuts as λ increases, until the bounds of the other MEBs have been reached.

The second vector in both groups (3 and 8) represents the allocation of manpower in AFMC during a large acquisition period, whether it be for a single system or multiple systems. During such an acquisition period, the primary MEBs of importance would be PM, S&T, and T&E while S&IO would not be regarded as important [7]. Therefore, PM, S&T, and T&E were given positive values of 1 while S&IO was given the value of 0. This value causes S&IO's objective function coefficient to decrease, in terms of relative importance, while the other coefficients grow larger (except for BOS in the first group), representing the growing importance of the acquisition-related MEBs. Then as λ increases, S&IO would absorb more of the required reductions for the group in which

29

BOS had a value of 10. In the group where BOS had a value of 0, BOS and S&IO would take the majority of reductions.

The third vector in both groups (4 and 9) is similar to direction vectors (2) and (7), respectively; however, PM is given more importance by doubling the rate at which it incrementally increases. The same results would be expected as above but now, as λ increases, PM would take even fewer cuts, if any at all, independent of which group the vector is in. This could be indicative of an acquisition period in which the majority of testing and scientific study has been accomplished and the actual purchase of the system is the key focus in the AFMC.

The final vector in both groups (5 and 0) contrasts the second vector in the groups. Direction vectors (5) and (0) represent periods when AFMC is concentrating largely on system maintenance. This normally occurs after a large acquisition period has concluded or during large system upgrades. In this case, S&IO becomes the important MEB while PM, S&T, and T&E are less important. Now, as λ increases, S&IO would be expected to take fewer cuts regardless of the direction vector's group.

IV. Analysis

4.1 Introduction

This chapter presents the results of the research. Included in the analysis are two initial states of objective function coefficients on which the eleven directional vectors are applied. Each of these combinations is performed twice where levels of the lower bound constraints differ. The initial results are then discussed, leading to further modifications and additional analysis in the parametric program.

4.2 Development of Initial Weighting Parameters

Since initial conditions and attitudes may vary greatly during the RA-IPT process, different scenarios were required to model the possible weighting schemes configured by the MEB representatives. Various initial conditions were presented, but eventually these were narrowed down to two possible alternatives that could best reflect the possible attitudes in the RA-IPT process.

In order to better understand the relationships between the weightings given the MEBs, the objective function coefficients were normalized and forced to sum to one for each year's optimization within each iteration. This would cause the objective function coefficients to become the percentage of weight assigned to each MEB during each year of the planning horizon. For example, an objective function coefficient vector of [0.667 0.08325 0.08325 0.08325 0.08325], given the order [BOS PM S&IO S&T T&E], would indicate that the decision makers assigned twice as much weight, or

31

importance, to the BOS MEB as all the others combined. It would also indicate that the

PM, S&IO, S&T, and T&E MEBs were all considered of equal importance in this specific

instance.

4.2.1 Initial State 1. The first set of initial objective function coefficients was

derived from the default weights Bishop previously adopted in ESP. These default

weights represented the current percentage of manpower allocations that each specific

MEB was assigned out of the total possible manpower allocations throughout AFMC for

each year [1:27]. This set of initial objective function coefficients can be seen in Table 4.1

and is derived from the data previously listed in Table 3.1. Since they represent the

current percentage of manpower assigned to each MEB, they are already normalized for

all five years in the first iteration. The normalization is also performed throughout the five

optimizations in the remaining iterations after each recommended cut for a given year has

been applied to the manpower levels.

Table 4.1 State 1 Initial Objective Function Coefficient Values

MEB	Year 1	Year 2	Year 3	Year 4	Year 5
BOS	.30728	.31411	.32099	.32169	.32217
PM	.27629	.27245	.27830	.27721	.27685
S&IO	.28430	.27692	.26464	.26488	.26437
S&T	.05979	.06164	.06184	.06218	.06264
T& E	.07475	.07487	.07423	.07404	.07396

By using these current percentages as objective function coefficient values, it is

implied that the larger an MEB is, compared to another MEB, the greater its importance

to the RA-IPT members.[5] This might be the case if monetary allocations to each MEB, out of the AFMC budget, were directly proportional to the amount of manpower in each MEB. With these initial objective function coefficient values, it would be expected that in the linear programming problem, the larger MEBs (BOS, S&IO, and PM) would take proportionally fewer manpower reductions than the smaller MEBs (S&T and T&E).

4.2.2 Initial State 2. The second set of initial objective function coefficient values was designed to examine the opposite effect of the previous concept. In this set, the *reciprocal* of the current percentage of manpower allocations that a specific MEB maintains becomes its initial objective function coefficient value. These values, after being normalized, are presented below in Table 4.2.

Table 4.2 State 2 Initial Objective Function Coefficient Values

MEB	Year 1	Year 2	Year 3	Year 4	Year 5
BOS	0.080366	0.079501	0.077632	0.077560	0.077631
PM	0.089381	0.091657	0.089541	0.090005	0.090339
S&IO	0.086862	0.090178	0.094163	0.094194	0.094603
S&T	0.413024	0.405127	0.402962	0.401258	0.399269
T& E	0.330367	0.333537	0.335702	0.336983	0.338159

Now, the larger an MEB is in relation to another, the less important it is to the decision maker. Smaller MEBs, specifically S&T and T&E, have the advantage in this scenario and would be expected to have proportionally and absolutely fewer reductions in the solution of the linear programming problem. This might be the case if manpower

---

[5] Importance means that the decision maker is less likely to reduce manpower allocations from the specific MEB which he or she finds more important.

reductions are more harmful for a small MEB because each cut removes a larger

percentage of the organization's mission capability. This infers that the larger MEBs

would be able to absorb the reductions better since they have a larger initial force.

4.3 Initial Detail of Runs Performed

The initial results consist of analysis done on forty-four runs of the FORTRAN

program. These runs are based on the combinations of the two initial objective function

coefficient vectors with the eleven direction vectors where λ varies from 0 to 1 by scalar

step sizes of 0.01. Twenty-two of these combinations contain the lower bound constraints

set at 97% and are referred to as group A, while the other twenty-two contain the lower

bound constraints set at the 95% level and are referred to as group B. The initial cuts

were determined based on manpower reductions of 2000 every year over the five year

horizon. (A cut of 2000 per year represents the current trend in manpower reductions at

AFMC [7].) The results are presented by initial objective function coefficient vectors

within the two groups of runs.

4.4 Deciphering the Output

The analysis concentrated on the *location* of the changes in the manpower allocation

values of the decision variables over the range of λ. The locations of these changes are

represented in the results by the actual iteration number where they take place and are

referred to as points of interest. The output was transformed into graphs indicating the

manpower allocation values for each MEB over the five year horizon. This allows the points of interest to be determined for each individual MEB.

Examples of output from one of the more interesting set of results from Group A, Run 20, are shown in the following figures. In this run, the initial objective function coefficients are the reciprocals of the current percentage of manpower associated with each MEB (State 2) and the lower bound constraints are set at 97%. The direction vector for Run 20 was [10 0 1 0 0] with an MEB order of: BOS, PM, S&IO, S&T, T&E.

The first graph, Figure 4.1, indicates that changes in the manpower allocation are taking place for the BOS decision variable at distinct points of interest within the range of λ. Specifically, at iterations 1, 2, 3, and 4, where λ respectively equals 0.01, 0.02, 0.03, and 0.04, the size of manpower reductions decreased for the BOS MEB during every year. This would be expected since BOS had the most positive rate of change in the direction vector, with a value of 10, for this run.

BOS
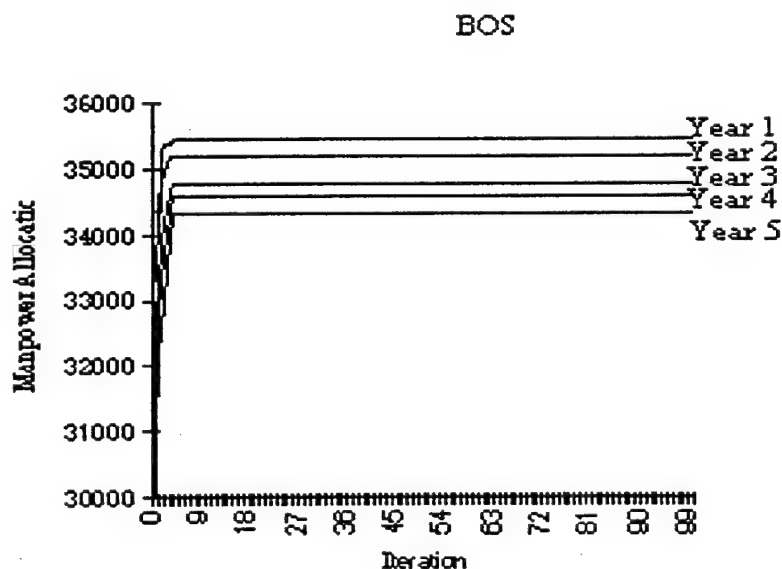
Figure 4.1 Example of Run 20's Output Changes in BOS's Manpower Allocation

35

The next graph, Figure 4.2, shows that the manpower allocations also change for the PM MEB over the range of λ. In this case, the specific point of interest occurs at iteration 1, where λ equals 0.01. At this point, the manpower allocated to PM drops, which is to be expected since PM has one of the lowest rates of change in the direction vector, with a value of 0, used for the example.

PM
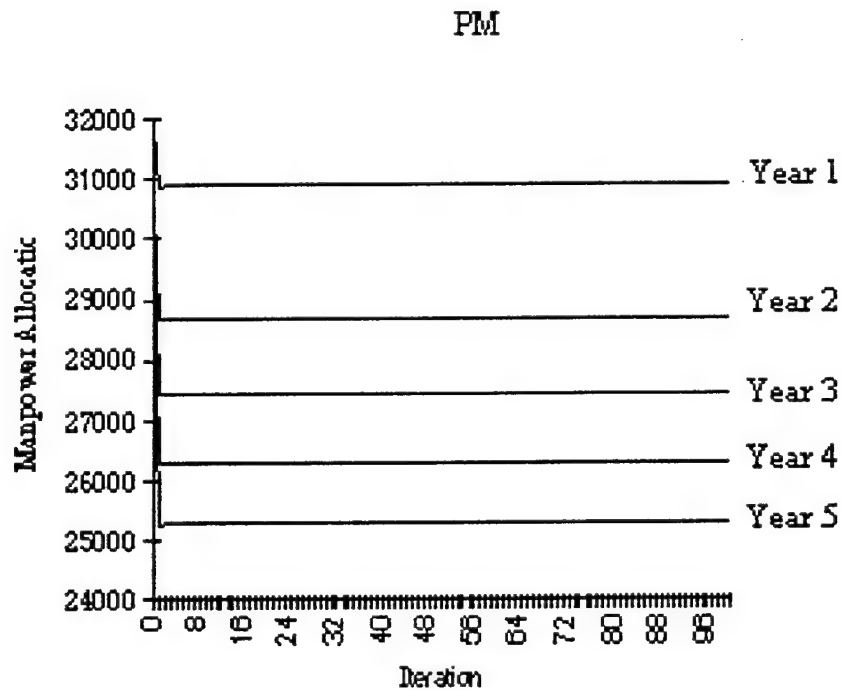
Figure 4.2 Example of Run 20's Output Changes in PM's Manpower Allocation

Figure 4.3 presents an interesting case. S&IO has the second highest positive rate of change in the direction vector, yet it loses manpower allocations at iteration 2 after initially gaining manpower at iteration 1. It then gains manpower allocations again at iterations 25, 27, 33, and 35. This is an indication of the interaction occurring between

36

the initial objective function coefficient and the rate of change. While S&IO has the
second highest rate of change in the direction vector, it has the second lowest initial
objective function coefficient.

S&IO
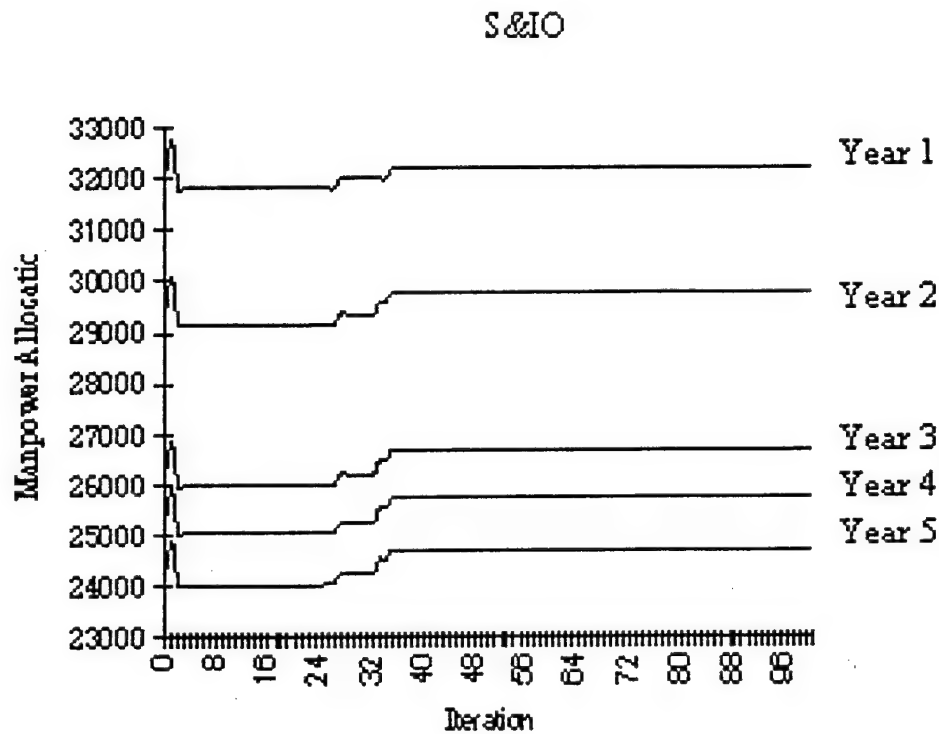
Figure 4.3 Example of Run 20's Output Changes in S&IO's Manpower Allocation

In both Figures 4.4 and 4.5, S&T and T&E take additional manpower reductions as
λ is increased over its range. S&T takes additional reductions at iterations 4, 33, and 35,
while T&E takes additional reductions at iterations 3, 4, 25, and 27. This would be as
expected since both of these MEBs started the run with the highest values in the initial
state of the objective function coefficients and had the lowest values in the direction
vector.

37

S&T
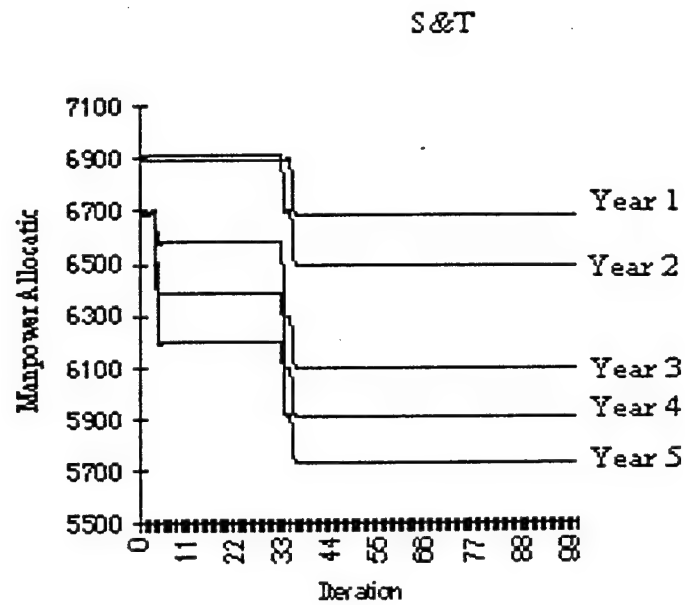
Figure 4.4 Example of Run 20's Output Changes in S&T's Manpower Allocation

T&E
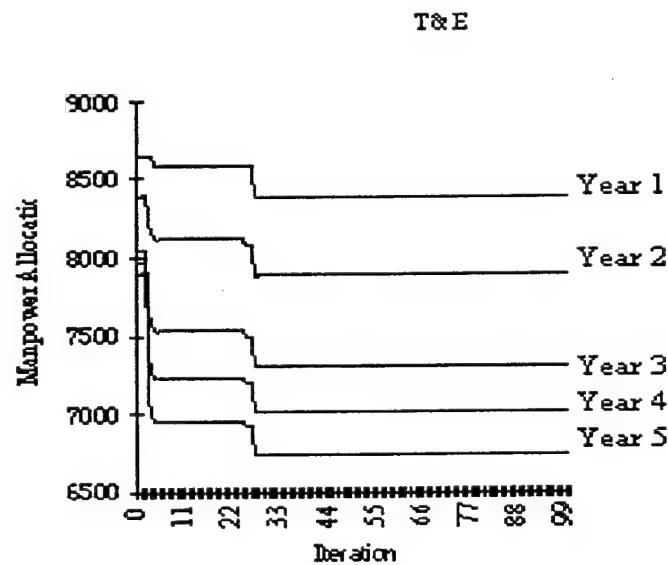
Figure 4.5 Example of Run 20's Output Changes in T&E's Manpower Allocation

In Appendix B, every point of interest (for all runs performed) is expanded to include the values of the objective function coefficients for each MEB. For each point of interest, the values listed are the average values of the objective function coefficients over the five optimizations. The average is taken since the changes could have occurred at any one or all of the five years within the iteration; therefore, the average gives the decision maker an understanding of the relative weight assigned to an MEB in that iteration without going into specific detail for each year in the iteration. The values are not the same throughout the five years since they are derived, at iteration 0, from the original manpower allocations made to the MEBs which differ throughout the years. Examples of the objective function coefficient values at points of interest for Run 20 are listed below. Table 4.3 lists the actual values used within each individual year's optimization while Table 4.4 lists the average value used over all five years.

In Table 4.4, S&T and T&E are assigned the most weight in the initial objective function, which corresponds to State 2 representing the smallest MEBs as the most important. As the iterations increase and λ incrementally becomes larger, it is evident that BOS increases in relative importance as the importance of PM, S&T, and T&E diminishes. However, it is interesting to see that S&IO's weight almost never changes while the remaining MEBs' weights do. Even with this pure coincidence, the results in Table 4.4 indicate that the direction vector [10 0 1 0 0] is properly functioning in this run. Relatively, S&IO does become more important than PM, S&T, and T&E while BOS becomes extremely important.

Table 4.3 Example of Objective Function Coefficient Values at Points of Interest
For Run 20

		Initial State:	2			Step Size:		0.01		
		Direction Vector:	0			Lower Bounds:		97%		
	ITERATION:	0	1	2	3	4	25	27	33	35
Year	LAMBDA:	0	0.01	0.02	0.03	0.04	0.25	0.27	0.33	0.35
1	BOS	0.080	0.162	0.230	0.286	0.334	0.688	0.700	0.730	0.738
	PM	0.089	0.081	0.073	0.067	0.062	0.024	0.023	0.019	0.018
	S&IO	0.087	0.087	0.088	0.088	0.088	0.090	0.090	0.090	0.090
	S&T	0.413	0.372	0.339	0.311	0.287	0.110	0.104	0.089	0.085
	T&E	0.330	0.298	0.271	0.248	0.229	0.088	0.083	0.071	0.068
2	BOS	0.080	0.162	0.229	0.285	0.333	0.688	0.700	0.730	0.738
	PM	0.092	0.083	0.075	0.069	0.064	0.024	0.023	0.020	0.019
	S&IO	0.090	0.090	0.090	0.090	0.090	0.091	0.091	0.091	0.091
	S&T	0.405	0.365	0.332	0.305	0.281	0.108	0.102	0.088	0.084
	T&E	0.334	0.300	0.273	0.251	0.232	0.089	0.084	0.072	0.069
3	BOS	0.078	0.160	0.228	0.284	0.332	0.687	0.700	0.730	0.738
	PM	0.090	0.081	0.073	0.067	0.062	0.024	0.023	0.019	0.018
	S&IO	0.094	0.094	0.094	0.093	0.093	0.092	0.092	0.092	0.092
	S&T	0.403	0.363	0.330	0.303	0.280	0.107	0.102	0.087	0.083
	T&E	0.336	0.302	0.275	0.252	0.233	0.090	0.085	0.073	0.069
4	BOS	0.078	0.160	0.228	0.284	0.332	0.687	0.700	0.729	0.738
	PM	0.090	0.081	0.074	0.068	0.063	0.024	0.023	0.019	0.019
	S&IO	0.094	0.094	0.094	0.093	0.093	0.092	0.092	0.092	0.092
	S&T	0.401	0.361	0.329	0.302	0.279	0.107	0.101	0.087	0.083
	T&E	0.337	0.304	0.276	0.253	0.234	0.090	0.085	0.073	0.069
5	BOS	0.078	0.160	0.228	0.284	0.332	0.687	0.700	0.730	0.738
	PM	0.090	0.081	0.074	0.068	0.063	0.024	0.023	0.020	0.019
	S&IO	0.095	0.094	0.094	0.094	0.093	0.092	0.092	0.092	0.092
	S&T	0.399	0.360	0.327	0.300	0.277	0.106	0.101	0.086	0.082
	T&E	0.338	0.305	0.277	0.254	0.235	0.090	0.085	0.073	0.070

Table 4.4 Example of Average Objective Function Coefficient Values at Points of Interest
For Run 20

	Initial State:	2			Step Size:		0.01		
	Direction Vector:	0			Lower Bounds:		97%		
	ITERATION:	0	1	2	3	4	25	27	33	35
	LAMBDA:	0	0.01	0.02	0.03	0.04	0.25	0.27	0.33	0.35
BOS		0.079	0.161	0.228	0.285	0.332	0.688	0.700	0.730	0.738
PM		0.090	0.081	0.074	0.068	0.063	0.024	0.023	0.019	0.019
S&IO		0.092	0.092	0.092	0.092	0.092	0.091	0.091	0.091	0.091
S&T		0.404	0.364	0.331	0.304	0.281	0.108	0.102	0.087	0.083
T&E		0.335	0.302	0.275	0.252	0.233	0.089	0.084	0.072	0.069

4.5 Group A Results

This first group of runs, with lower bounds constrained at the 97% level, set a template for the results that were discovered in this research. A majority of the runs presented some evidence of change in the decision variables throughout the range of λ, while about one-fourth of the runs showed no sensitivity to the multiple changes in objective function coefficient values. In the following tables, each run is referenced by a run code. The first number in this code represents the state that the run initiated from while the following number indicates the direction vector applied in the run. Run 20 would imply that direction vector 0 was used in conjunction with the initial State 2.

4.5.1 State 1 Results. The State 1 results, listed in Table 4.5, indicate that both sets of runs, including direction vectors with a BOS value of 0 (direction vectors 2, 3, 4, and 5) and a value of 10 (direction vectors 7, 8, 9, and 0), had significant changes in the decision variables over the range of λ. The number of runs with changes in the first few iteration steps indicates that the linear program is extremely sensitive to the initial objective function coefficient values.

Table 4.5 Iterations with Changes in Decision Variables for Each MEB from State 1, Group A

Run Code	BOS	PM	S&IO	S&T	T&E
10	-	2	2	-	-
11	-	-	-	-	-
12	3,4,6	3,4	3,4,6	-	-
13	4,5,24,25,26	1,4,5	1	25,26	24,26
14	2,3,24,25,26	1,2,3	1	25,26	24,25,26
15	3,4,6	2	2,3,4,6	-	-
16	-	1	1	-	-
17	-	-	-	-	-
18	3,5,7	5,7	3,5,7	-	-
19	-	1	1	-	-
111	4,5,7,8,9	4,5,8,9,10,11	1	7,8	8,9,10,11

41

4.5.2 State 2 Results. The set of results listed in Table 4.6 indicates that changes

are taking place in a majority of the runs. Patterns can be seen emerging from the two sets

of runs with contrasting values for the BOS MEB. Since BOS is the least important in the

initial state of the objective function coefficients, it begins every scenario by taking the

most reductions. When BOS is important in the direction vector (direction vectors 6, 7, 8,

9, and 0) driving the scenario, it takes fewer reductions within the first few iterations of

the run. When BOS is not important in the direction vector, its manpower allocations do

not change over the range of λ.

Table 4.6 Iterations with Changes in Decision Variables for Each MEB from
State 2, Group A

Run Code	BOS	PM	S&IO	S&T	T&E
20	1,2,3,4	1	1,2,25,27,33,35	4,33,35	3,4,25,27
21	-	-	-	-	-
22	-	-	-	-	-
23	-	-	-	-	-
24	-	25,33	-	33	25
25	-	1	1,25,33	33	25
26	1,9,17	1	1,17,33	17,33	9,13
27	1,3,4	1	1	4	3
28	1,3,4	1	1	4	3
29	1,3,4	1,25,33	1	4,25,33	3,25
211	-	-	-	-	-

4.6 Group B Results

This group of runs contained results similar to those in group A. However, the

change in the level of the constrained lower bounds, from 97% to 95%, was evident in

that a small number of the changes in the runs differed from Group A's.

4.6.1 State 1 Results.   The decision variables had values different from those in Group A in six of the eleven runs listed in Table 4.7. The most notable changes were the ones which occurred for the S&T and T&E MEBs in runs 14, 16, 18, and 19. These changes can be attributed to the fact that the larger MEBs can take larger cuts in this group of runs, thereby allowing the smaller MEBs to take fewer reductions.

Table 4.7 Iterations with Changes in Decision Variables for Each MEB from State 1, Group B

Run Code	BOS	PM	S&IO	S&T	T&E
10	-	2	2	-	-
11	-	-	-	-	-
12	4,5,6,24,25,26	4,5	6	25,26	24,25
13	5	1,5	1,21	25,26	21,25,26
14	3,25,26	1,3	1,3,21	1,21	25,26
15	6	2	2,6	-	-
16	-	1,11,21	1,11,21	11	21
17	-	-	-	-	-
18	-	1	1,21	-	21
19	-	1	1,21	-	21
111	5,7,8,9	1,5	1,6,7	6,7	7,8,9

4.6.2 State 2 Results.   Once again, the same patterns for BOS were shown in this set of runs, as listed below in Table 4.8. The difference in the lower bound constraints can be seen here by the fewer amount of points of interest found in Runs 20, 24, 25, 26, 27, 28, 29, and 211, as compared to those found in Table 4.6. Specifically, in Runs 24, 25, 27, and 28, S&T and T&E no longer take any additional reductions since the change in the lower bound now allows the remaining MEBs to take larger amounts of reductions.

Table 4.8 Iterations with Changes in Decision Variables for Each MEB from State 2, Group B

Run Code	BOS	PM	S&IO	S&T	T&E
20	1	1	1,25,33	33	25
21	-	-	-	-	-
22	-	-	-	-	-
23	-	-	-	-	-
24	-	-	-	-	-
25	-	1	1	-	-
26	1	1	1,13,33	33	13
27	1	1	1	-	-
28	1	1	1	-	-
29	1	1,25,33	1	33	25
211	-	-	-	-	-

4.7 Duplicated Runs

In the initial FORTRAN code, each direction vector was multiplied by an incremental step size of 0.01 for 100 iterations, giving λ an initial range of (0,1). A different step size was chosen for each of the two different initial states to determine if other important changes occurred inside or outside the initial range of λ. For State 1, an incremental step size of 0.1 was chosen, giving λ a new range of (0,10). For State 2, an incremental step size of 0.003 was chosen, giving λ a new range of (0,0.3).

4.8 Group A Results Using Different λ

4.8.1 State 1 Results.   Moving to the larger step size of $\lambda = 0.1$ did not produce additional results for this case. Changes at the single-digit iterations presented in Table 4.9 reflect the same changes as seen in the previous Group A State 1 results (Table 4.5).

Table 4.9 Iterations with Changes in Decision Variables for Each MEB from
State 1, Group A with $\lambda = 0.1$

Run Code	BOS	PM	S&IO	S&T	T&E
10	-	1	1	-	-
11	-	-	-	-	-
12	1	1	1	-	-
13	1,3	1	1	3	3
14	1,3	1	1	3	3
15	1	1	1	-	-
16	-	1	1	-	-
17	-	-	-	-	-
18	-	1	1	-	-
19	-	1	1	-	-
111	1,2	1,2	1	1	2

4.8.2 State 2 Results. The results for this case indicate that the change in step size was not very illuminating. Only a few additional points of interest were found while previous points were verified. The addition of the points of interest added detail to points of interest listed in Table 4.6.

Table 4.10 Iterations with Changes in Decision Variables for Each MEB from
State 2, Group A with $\lambda = 0.003$

Run Code	BOS	PM	S&IO	S&T	T&E
20	1,4,9,12	1,4	1,4,82,87	12	12,82,87
21	-	-	-	-	-
22	-	-	-	-	-
23	-	-	-	-	-
24	-	81	-	-	81
25	-	1	1,82	-	82
26	1,3,28,56	1	1,3,41	56	28,41
27	1,10,13	1	1	13	10
28	1,10,13	1	1	13	10
29	1,10,13	1,81	1	13	10,81
211	-	-	-	-	-

4.9 Group B Results Using Different λ

4.9.1 State 1 Results.
Once again, enlarging the step size to 0.1 did nothing more than reflect the results presented in Table 4.7. No additional points of interest were found in this particular set of runs.

Table 4.11 Iterations with Changes in Decision Variables for Each MEB from State 1, Group B with λ = 0.1

Run Code	BOS	PM	S&IO	S&T	T&E
10	-	1	1	-	-
11	-	-	-	-	-
12	1,3	1	.1	3	3
13	1,3	1	1,3	3	3
14	1,3	1	1,3	3	3
15	1	1	1	-	-
16	-	1,2,3	1	2	3
17	-	-	-	-	-
18	-	1	1,3	-	3
19	-	1	1,3	-	3
111	1,2	1	1	1	1,2

4.9.2 State 2 Results.
Decreasing the step size to 0.003 did nothing more than reflect the previous points of interest found in Table 4.8, as seen listed in Table 4.12.

Table 4.12 Iterations with Changes in Decision Variables for Each MEB from State 2, Group B with λ = 0.003

Run Code	BOS	PM	S&IO	S&T	T&E
20	1	1	1,82	-	82
21	-	-	-	-	-
22	-	-	-	-	-
23	-	-	-	-	-
24	-	-	-	-	-
25	-	1	1	-	-
26	1,3	1	1,3,41	-	41
27	1	1	1	-	-
28	1	1	1	-	-
29	1	1,81	1	-	81
211	-	-	-	-	-

4.10 Direction Vector Analysis

It is important to the RA-IPT decision makers to know where and why the decision variables change. The answer to why comes from the combination of the initial objective function coefficient, the direction vector, and the constraints. Since changes caused by the constraints and bounds are hard to determine, the focus of this section is on the direction vectors.

Direction vector (1) was the test case and its results performed as expected. Direction vector (6) represented a possible ranking scheme for the MEBs and produced some of the most interesting results. These results depended a great deal on the initial objective function coefficient values and λ. With State 1, this vector would shift manpower allocations from PM to S&IO, S&T, and T&E, as expected. With State 2, this vector would greatly increase the manpower allocations to the BOS while decreasing PM. Yet, it would then cause S&IO to flip between decreasing and increasing manpower allocations with BOS, T&E, and S&T, depending on the size of λ.

The vector derived from the utility losses did not produce very interesting results. The majority of changes that occurred due to this vector happened within the first iterations, indicating that the solutions were very insensitive to the combinations of λ and this vector. In State 2, this direction vector never produced any changes in manpower allocations. This can be explained by the fact that in State 2, the smaller MEBs were greatly favored over the larger in both the initial objective function coefficient values and in the direction vector derived from Smith's work.

47

The large acquisition vectors with BOS unimportant (3 and 4) performed as expected, showing few, but noticeable changes. Differences existed between the combinations of these vectors and the initial states. In State 1, both BOS and S&IO took increased reductions, while in State 2, neither took additional reductions beyond the initial iteration. The large acquisition vectors with BOS important (8 and 9) had contrasting results, as expected. In State 1, these vectors had S&IO increasing its reductions alone, and then sharing the increase in reductions with PM, S&T, and T&E in State 2.

The system maintenance vectors (5 and 0) yielded some interesting results. In State 1, these vectors took fewer reductions from the S&IO MEB and more reductions from the PM MEB and occasionally T&E as λ increased. But, when BOS was important in State 2, S&IO would actually take additional reductions at first to help balance out the fewer reductions allocated to BOS. It would then take fewer reductions during later iterations.

In the contrasting BOS vectors (2,3,4,5 v 7,8,9,0), the state was an important factor in determining if BOS would take more or fewer reductions. For State 1, every run with either vector 2, 3, 4, or 5 had BOS taking greater reductions while runs with either vectors 7, 8, 9, or 0 had no changes in BOS. The opposite was true for State 2. Vectors 2, 3, 4, and 5 produced runs with no changes in BOS while vectors 7, 8, 9, and 0 produced runs with changes in BOS.

4.11 MEB Analysis

With manpower allocations being so valuable to each MEB, understanding the points of change through the eyes of an MEB member can be crucial to the setting of

relative weights in the RA-IPT process. During an RA-IPT process, each MEB will present its case on why it should be weighted as the most important and therefore receive the fewest reductions. Throughout the process, MEBs will continue to compete for relative weighting as the process continues and the weightings shift among the MEBs. The results of this analysis indicate that during much of the process, the manpower allocations remain unchanged for specific MEBs once certain levels of importance are reached given an initial state and specific scenarios (represented by the different direction vectors).

4.11.1 BOS Results. In the runs initiated with State 1, BOS was given 31.7% of the weight to start with. With such a large percentage of the weight assigned to this MEB, it never took additional manpower reductions in any of the scenarios in which its relative weight increased above 32%. In fact, during the system maintenance scenario, BOS received fewer manpower reductions when it achieved 43%, 50%, and 53% of the weight. However, in all of the scenarios involving BOS losing its relative weight, it did take additional reductions when it began to fall below 30% of the total weight. The increase in reductions would then halt around the 28% level until it fell down around the 16%-18% level where it would drop for the last time in the scenarios. When the scenario based on the utility values was applied to this State, regardless of the lower bound setting, BOS received additional reductions as its weight decreased to 24.2%, 22.5%, 21.7%, 21.1%, 20.46%, and 16.7%.

In the runs initiated with State 2, BOS was given only 7.9% of the weight to start with. With the lowest weighting, it never took additional reductions in any of the

scenarios since it was already taking the most reductions it could given the constraints of the problem. However, its manpower allocations during acquisition scenarios would increase when its weight would increase to around 10%, 27%, and 30%. During the system maintenance scenario with the lower bound set to 97%, BOS had increases in manpower allocations when its weight reached 10.5%, 17.5%, and 31.3%. In the same scenario with the lower bound set at 95%, BOS only saw increases in manpower allocations when its weight reached 10.5%. Using the ranking scenario with a lower bound set at 97%, BOS was found to have increases in manpower allocations when its weight reached 9%, 10.9%, 22.1%, and 26.1%. When the lower bound was set to 95% however, BOS only had increases at weight settings of 9% and 10.9%.

4.11.2 PM Results. In the runs initiated with State 1, PM was given 27.6% of the weight to begin with. During the acquisition scenarios for this state, PM increased in manpower allocations immediately within the addition of only two more percentage points. By the time it held at least 30% of the weight, it never took any of the manpower reductions. However, in one interesting case with the acquisition scenario, PM actually took additional manpower reductions to compensate for the BOS weight increasing drastically. In this scenario, PM's weight reached 20% and 18% when it took the additional reductions. In the system maintenance scenarios, PM would take additional reductions when its weight fell to 22%, but no additional reductions after that. In the scenarios that included the linear ranking of the MEBs, PM took additional reductions away from the S&IO MEB when PM's weight fell to 25% and the lower bound was set at 97%. When the lower bound was set to 95%, PM took additional reductions away from

50

S&IO, S&T, and T&E when its weight fell to 25%, 15%, and 12%. When the scenario based on the utility values was applied to this state and the lower bound constraints set at 97%, PM received fewer reductions as its weight decreased to 24.1%, 23.5%, 22.5%, 22.1%, 21.7%, 21.4%, 21.1%, 15.6, and 11.8%. However, when the lower bound constraints were set at 95%, PM received fewer reductions as its weight decreased to 26.5%, 23.5%, and 21.4%. The fewer reductions, given the decreasing weights, can be explained by PM having a relatively larger value in the direction vector, than BOS and S&IO, who therefore, end up taking the reductions away from PM.

In the runs initiated with State 2, PM was given only 9% of the weight to begin with and changes in manpower allocations were extremely sensitive to changes in this relative weight: often, changes of only 0.1% reflected changes in the reductions. When BOS was important in the acquisition scenarios, PM took additional reductions when its weight dropped less than 0.4% to compensate for additional BOS manpower allocations. (Even though the direction vectors associated with the acquisition period gave PM an increasing value, PM's relative weight decreased throughout the runs since BOS's increased at a much greater rate.) The only scenario in which PM had fewer reductions was during the acquisition period with PM growing twice as fast as S&T and T&E. In this run, PM took an additional reduction when its weight reached 9.2% but then took fewer reductions when its weight reached 13.1%.

4.11.3 S&IO Results. In the runs initiated with State 1, S&IO was given 27% of the weight to begin with. During acquisition scenarios (in which S&IO had a value of 0 in the direction vectors) and BOS was not important, S&IO took additional reductions when

51

its weight dropped to 26%, 24%, 17%, and 14%. When BOS was important in the acquisition scenarios and the lower bound constraints were set at 95%, S&IO took additional reductions when its weights dropped to 24% and 7%. When BOS was important in the acquisition scenarios and the lower bound constraints were set at 97%, S&IO took additional reductions when its weights dropped to 24%, 19.5%, 16%, and 14%. When BOS was important during a system maintenance scenario, S&IO's reductions decreased when its weights reached 24%. Again, the relative weight dropped even though S&IO had a positive value in the direction vector for this scenario because BOS's relative weight increased at a faster pace. The point at which S&IO's weight decreased to 24% is also where S&IO's weight becomes larger than PM's (23%) and therefore the reductions are decreased for S&IO while increasing for PM. When BOS was not important during a system maintenance scenario, S&IO's reductions decreased when its weight reached 28.5%, 29%, 30%, and 31%. In the scenarios that were based on the linear rankings, additional reductions were made only as S&IO's weight dropped only 0.5%. When the scenario based on the utility values was applied to this state and the lower bound constraints were set at 97%, S&IO received more reductions as its weight decreased to 15.6%. However, when the lower bound constraints were set at 95%, S&IO received more reductions as its weight decreased to 25%, 18.5%, and 17.7%.

In the runs initiated with State 2, S&IO was given a starting weight of 9.2%. When BOS was important in the acquisition scenarios, S&IO took additional reductions when its weight dropped to around 8.9% and 8.1%. When system maintenance drove the direction vectors and BOS was unimportant, S&IO took fewer manpower reductions when its

52

relative weight increased to 10.1% and the lower bound was set at 95%. When its lower bound was set to 97% under the same scenario, S&IO had fewer reductions taken as its weight reached 10.1%, 27.4%, and 31.7%. When system maintenance drove the direction vectors and BOS was important, S&IO actually took both increasing and decreasing manpower reductions while its relative weight stayed around 9.2%. This was due to the extreme changes in the remaining MEBs' weightings. When the scenarios included the linear ranking of the MEBs, S&IO took additional reductions when its weight reached 11.5%, but took fewer reductions as its weight reached 20.7% and 23.4%.

4.11.4 S&T Results. In the runs initiated with State 1, S&T was given an initial weight of 6.1%. During acquisition periods in which BOS was not important and PM increased as fast as S&T (same value in the direction vector), S&T took fewer reductions when its weight reached 17.8% and 18%. The same was true during acquisition periods in which: BOS was not important, PM increased faster than S&T (PM had a higher value in the direction vector), and the lower bound constraints were set to 97%. However, S&T took fewer reductions when its weight reached 6.9% and 14.8% during acquisition periods in which: BOS was not important, PM increased faster than S&T (PM had a higher value in the direction vector), and the lower bound constraints were set to 95%. When the lower bound constraints were set to 95% and the direction vector using linear ranking was applied, S&T took fewer reductions when its weight reached 14.8%. When the scenario based on the utility values was applied to this state and the lower bound constraints were set at 97%, S&T received fewer reductions as its weight increased to

20.6% and 21.7%. However, when the lower bound constraints were set at 95%, S&T received fewer reductions as its weight increased to 19.3% and 20.6%.

In the runs initiated with State 2, S&T was given the highest initial value of 40.4%. Additional reductions in S&T's manpower were only found in three runs in which the lower bound was set to 95%, all of which included BOS as important. At this setting in the system maintenance scenario, S&T took additional reductions when its weight dropped to 8.7%. At this lower bound setting in the linearly ranked scenario, S&T took additional reductions when its weight reached 23.4%. At the 95% lower bound setting in the acquisition scenario with a strong PM, S&T took additional manpower reductions when its weight reached 13.1%. When the lower bound constraint was set to 97%, more changes in manpower allocation were found. At this lower bound setting in the system maintenance scenarios with BOS important, S&T took more reductions when its weight reached 28.1%, 8.7%, and 8.3%. In this same scenario with BOS unimportant, S&T took additional reductions when its weight reached 30.4%. During acquisition scenarios in which BOS was important, S&T took additional reductions when its weight reached around 29%. In the linearly ranked scenario, S&T took additional manpower reductions when it weight reached 25.8% and 23.4%.

4.11.5 T&E Results. In the runs initiated with State 1, T&E was given an initial weight of 7.4%. Indifferent to the lower bound setting, T&E had fewer reductions occur in the acquisition scenarios and BOS not important when its weight increased to around 15.5%. However, when the lower bound constraints were set to 95%, T&E also received fewer reductions during scenarios with acquisition periods and BOS important when its

weight increased to around 7.5%. At this lower bound setting, T&E also had fewer

reductions occur when its weight increased to 12% in the scenario involving the linear

rankings. When the scenario based on the utility values was applied to this state and the

lower bound constraints were set at 97%, T&E received fewer reductions as its weight

increased to 17.5%, 18.2%, 18.8%, and 19.4%. However, when the lower bound

constraints were set at 95%, T&E received fewer reductions as its weight increased to

16.8%, 17.5%, and 18.2%.

In the runs initiated with State 2, T&E was given an initial weight of 33.5% and had

almost identical results as S&T. Additional reductions in T&E's manpower were only

found in three runs in which the lower bound was set to 95%, all of which included BOS

as important. At this setting in the system maintenance scenario, T&E took additional

reductions when its weight dropped to 8.9%. At this lower bound setting in the linearly

ranked scenario, T&E took additional reductions when its weight reached 26.9%. At the

95% lower bound setting in the acquisition scenario with a strong PM, T&E took

additional manpower reductions when its weight reached 13%. When the lower bound

constraint was set to 97%, more changes in manpower allocation were found. At this

lower bound setting in the system maintenance scenarios with BOS important, T&E took

more reductions when its weight reached 25.2%, 23.3%, 8.9% and 8.4%. In this same

scenario with BOS unimportant, T&E took additional reductions when its weight reached

33.2%. During acquisition scenarios in which BOS was important, T&E took additional

reductions when its weight reached around 26%. In the linearly ranked scenario, T&E

took additional manpower reductions when it weight reached 21.9% and 20.2%.

4.12 Analytical Conclusions

The analysis has shown that efforts within the RA-IPT process can be made more efficient. MEB decision makers can view the points of interest associated with MEBs during the different scenarios as points of concern: where their respective weight does matter to their manpower allocation. Each MEB now has the ability to determine the benefits of pursuing the allocation of additional weight. The results have shown that sometimes the addition of a small amount of weight can be beneficial, while at other times, extremely large amounts of weight might not be beneficial to an MEB. They have also shown that there are times in which an MEB's manpower allocation relies strongly on the weights of the other MEBs. For example, in Figure 4.3 and Table 4.4, S&IO had its weight jump between 9.1% to 9.2%, but its manpower reductions increased and decreased depending on the weights assigned to the remaining MEBs. Given this scenario in an RA-IPT process, S&IO would know not to fight for changes in its own weight; rather, the changing of weights assigned to other MEBs are more important to S&IO. These results give the MEB decision makers a "road map" to the RA-IPT process by allowing them to spot the possible trouble areas in the weighting scheme where compromise will be needed as well as detailing the areas where conflicts can be avoided due to unchanging manpower allocations.

V. Conclusions and Recommendations

5.1 Conclusions

The objective of this research was to perform analysis that would allow the AFMC decision makers to better understand the effects of their inputs into the ESP model. The research concludes that the analysis presented can provide a better understanding of the ESP results and allow for a more effective and efficient RA-IPT process. This research provided two items of importance to the RA-IPT leaders: analysis completed on the effects of multiple simultaneous changes in the weighting schemes of the MEBs and a Microsoft EXCEL spreadsheet program to analyze other possible weighting schemes that might be derived during an RA-IPT process.

The analysis presented was able to provide the valuable insights that could be used during the RA-IPT process. Specific points of interest were shown to change the optimal manpower allocations and indicate where certain MEBs might benefit by increasing their relative weight. Ranges between these points were indicative of unchanging solutions and therefore, no benefit to MEBs for increasing their relative weights at these points. Based on the structure of the RA-IPT process, this analysis is ideal for the top-level MEB decision makers who are mainly concerned with the manpower reductions at their level. (Lower levels in the hierarchy, including the centers and two-letter organizations, can be fully analyzed with ESP *after* the MEB level is resolved.) MEB decision makers can now use this information in the RA-IPT to determine when compromise is necessary and when

it is not. Time and efforts can be saved by knowing how changes in attitude will or will not effect the outcome of the manpower allocations in the RA-IPT process.

The program included with this research is based on a bounded knapsack problem and allows for quick-time responses to be easily accessible to RA-IPT decision makers. Its pictorial representation of the weighting schemes permits visual interpretations to be made by the decision maker, giving the decision maker a better understanding of his or her inputs into the problem. The analysis and the program, together, give a means to better solutions for the AFMC manpower problem. Adding ESP will give HQ AFMC a capable decision support system which could facilitate reducing manpower in the most mission effective way.

5.2 Recommendations

This research has provided analysis on simultaneous multiple changes in the objective function coefficients, representing changes in a relative weighting scheme for MEBs in the AFMC manpower allocation problem. The analysis provided the insight needed to make the RA-IPT process more effective, but areas still exist in which more work can be done.

Data on the sequence of events within the RA-IPT manpower reduction process was never researched. Insights into better model building could be gained by interviewing the MEB decision makers and the AFMC Commander. Smith's work on utility values could be incorporated more effectively into this research if the actual risk tolerances and utility

58

functions of manpower reductions from the leaders in the RA-IPT process could be modeled directly into the generation of the direction vectors.

Throughout the analysis, only three scalar step sizes were chosen. Results indicated that the majority of points of interest were derived from the 0.01 step size. Further research could perform the analysis on larger ranges of λ or on greater or smaller values of λ. However, by normalizing the objective function coefficients after incrementally increasing them, larger values of λ will eventually cause the objective function coefficient vectors to converge. (The same problem occurs with direction vectors that contain values more than an order of magnitude different from each other.)

The scope of this research focused solely on the MEB level. Parametric analysis could be performed down to the center and two-letter levels using the complete ESP model.

Another way to look at this problem is by restructuring the AFMC manpower allocation process. Smith investigates the idea of other possible structures of the AFMC hierarchical organization. For example, AFMC personnel could be divided into the program that their projects are most associated with (i.e. the F15 program, the A10 program, etc.). The programs could then be tied to a specific MEB and analyzed in that manner. However, the resources in the BOS MEB would be difficult to link directly to any one program, so it would have to remain as an entity outside the program structure.

Appendix A. Example of FORTRAN Code

```
c RUN26.FOR
c
c  The routines call intermediate C routines that call
c  the CPLEX library routines; this approach should work under most
c  runtime systems, even when there is no "interface" facility.
c  This approach does require using the CPLEX-supplied intermediate
c  C routines (INTERMED.C).
c
c  All of the intermediate routines are functions, and all arguments
c  are passed by reference.
c

      implicit none
      integer iteration

      do iteration = 0,100
        write (*,*) 'Iteration =',iteration
        CALL RUNCPLEX(iteration)
      end do

      stop

      end

c**************************************************************************
SUBROUTINE RUNCPLEX(iteration)

c  INTERFACE DEFINITIONS
      external slogfo !$pragma C (slogfo)
      external sscrin !$pragma C (sscrin)
      external sitfoi !$pragma C (sitfoi)
      external sitlim !$pragma C (sitlim)
      external iloadp !$pragma C (iloadp)
      external iopt   !$pragma C (iopt)
      external gmar   !$pragma C (gmar)
      external gmac   !$pragma C (gmac)
      external ilpwr  !$pragma C (ilpwr)
      external isolut !$pragma C (isolut)
      external iaddr  !$pragma C (iaddr)
      external icbds  !$pragma C (icbds)
      external slogfc !$pragma C (slogfc)
      external iccoef !$pragma C (iccoef)
      external iunlod !$pragma C (iunlod)

c  CPLEX CONSTANTS
      integer      mac
      parameter    (mac=5)
      integer      mar
      parameter    (mar=1)
      integer      macsz
      parameter    (macsz=7)
      integer      marsz
      parameter    (marsz=3)
      integer      matsz
      parameter    (matsz=7)
      integer      cstsz
      parameter    (cstsz=macsz*3+1)
      integer      rstsz
      parameter    (rstsz=marsz*3+1)
      integer      cex
      parameter    (cex=macsz-mac)
      integer      rex
      parameter    (rex=marsz-mar)
      integer      cnt
      parameter    (cnt=10)
```

60

```
      double precision INFBND
      parameter (INFBND = 1.0e+20)


c  PART I DECLARATIONS
      integer      objsen      / -1 /
      double precision objx(macsz)
     . / 0.30728, 0.27629, 0.28430, 0.05979, 0.07475,
     .   cex*0.0 /
      double precision rhsx(marsz)   / 113627, rex*0.0 /
      character*1     senx(marsz)   / 'E', rex*' ' /
      integer       matbeg(macsz) / 0, 1, 2, 3, 4, cex*0 /
      integer       matcnt(macsz) / 1, 1, 1, 1, 1, cex*0 /
      integer       matind(0:matsz-1)
     .        / 0, 0, 0, 0, 0, 2*0 /

c  ENTER VALUES OF THE CONSTRAINT MATRIX
      double precision  matval(0:matsz-1)
     .        / 1., 1., 1., 1., 1., 2*0.0 /
c  ENTER LOWER AND UPPER BOUNDS ON VARIABLES
      double precision  bdl(macsz)
     . / 28350,25491,26230,5516,6914, cex*0.0 /
      double precision  bdu(macsz)
     . / 35438, 31864, 32787, 6895, 8643, cex*0.0/
      character*3      datanm      / ' ' /
      character*3      objnm       / ' ' /
      character*3      rhsnm       / ' ' /
      character*3      rngnm       / ' ' /
      character*3      bndnm       / ' ' /
      character*3      cstore(macsz)
     .          / 'B1','P1','I1','S1','T1',cex*' ' /
      character*3      rstore(marsz) / 'c1', rex*' ' /
      character*3      estore(1)
      integer      idummy(1)
      double precision  ddummy(marsz)
      integer      namlen      / 3 /
      integer      lpstat
      double precision  obj
      double precision  x(macsz)
      double precision  pi(macsz)
      double precision  slack(macsz)
      double precision  dj(macsz)


c  LOCAL VARIABLES
      integer      status, j, cutsize
      integer      toosmall, toobig
      double precision  dc1a, dc2a, dc3a, dc4a, dc5a
      double precision  dc1b, dc2b, dc3b, dc4b, dc5b
      double precision  dc1c, dc2c, dc3c, dc4c, dc5c
      double precision  dc1d, dc2d, dc3d, dc4d, dc5d
      double precision  dc1e, dc2e, dc3e, dc4e, dc5e
      double precision  dctota,dctotb,dctotc,dctotd,dctote

      double precision  dub1a, dub2a, dub3a, dub4a, dub5a
      double precision  dub1b, dub2b, dub3b, dub4b, dub5b
      double precision  dub1c, dub2c, dub3c, dub4c, dub5c
      double precision  dub1d, dub2d, dub3d, dub4d, dub5d
      double precision  dub1e, dub2e, dub3e, dub4e, dub5e
      double precision  ub1a, ub2a, ub3a, ub4a, ub5a
      double precision  ub1b, ub2b, ub3b, ub4b, ub5b
      double precision  ub1c, ub2c, ub3c, ub4c, ub5c
      double precision  ub1d, ub2d, ub3d, ub4d, ub5d
      double precision  ub1e, ub2e, ub3e, ub4e, ub5e

      double precision  dlb1a, dlb2a, dlb3a, dlb4a, dlb5a
      double precision  dlb1b, dlb2b, dlb3b, dlb4b, dlb5b
      double precision  dlb1c, dlb2c, dlb3c, dlb4c, dlb5c
      double precision  dlb1d, dlb2d, dlb3d, dlb4d, dlb5d
      double precision  dlb1e, dlb2e, dlb3e, dlb4e, dlb5e
```

61

```fortran
      double precision drhsa, drhsb, drhsc, drhsd, drhse
      double precision totcut1,totcut2,totcut3,totcut4,totcut5
      integer      dv1, dv2, dv3, dv4, dv5
      integer      ind(cnt)
      character*1   lu(cnt)
      double precision bd(cnt)
      double precision prcnt

c FUNCTIONS
      integer      sscrin
      integer      slogfo
      integer      sitfoi
      integer      sitlim
      integer      iloadp
      integer      iopt
      integer      isolut
      integer      gmar
      integer      gmac
      integer      iaddr
      integer      icbds
      integer      ilpwr
      integer      slogfc
      integer      iccoef
      integer      iunlod

c INITIALIZE VARIABLES

      j = 0
      cutsize = 2000
      prcnt  = (97.0/100.0)
      totcut1 = 0.0
      totcut2 = 0.0
      totcut3 = 0.0
      totcut4 = 0.0
      totcut5 = 0.0
      dv1 = 5
      dv2 = 1
      dv3 = 4
      dv4 = 3
      dv5 = 2

c CREATE AND NORMALIZE DUMMY OBJECTIVE COEFFIECIENT VALUES FOR VARIABLE n (1-5) AND
c YEARS 1-5 (a,b,c,d,e)
      dc1a = (1.0 / 0.30728) + dv1*(iteration * .01)
      dc2a = (1.0 / 0.27629) + dv2*(iteration * .01)
      dc3a = (1.0 / 0.28430) + dv3*(iteration * .01)
      dc4a = (1.0 / 0.05979) + dv4*(iteration * .01)
      dc5a = (1.0 / 0.07475) + dv5*(iteration * .01)
      dctota = dc1a+dc2a+dc3a+dc4a+dc5a
      dc1a = dc1a/dctota
      dc2a = dc2a/dctota
      dc3a = dc3a/dctota
      dc4a = dc4a/dctota
      dc5a = dc5a/dctota

      dc1b = (1.0 / 0.31411) + dv1*(iteration * .01)
      dc2b = (1.0 / 0.27245) + dv2*(iteration * .01)
      dc3b = (1.0 / 0.27692) + dv3*(iteration * .01)
      dc4b = (1.0 / 0.06164) + dv4*(iteration * .01)
      dc5b = (1.0 / 0.07487) + dv5*(iteration * .01)
      dctotb = dc1b+dc2b+dc3b+dc4b+dc5b
      dc1b = dc1b/dctotb
      dc2b = dc2b/dctotb
      dc3b = dc3b/dctotb
      dc4b = dc4b/dctotb
      dc5b = dc5b/dctotb

      dc1c = (1.0 / 0.32099) + dv1*(iteration * .01)
```

```
        dc2c = (1.0 / 0.27830) + dv2*(iteration * .01)
        dc3c = (1.0 / 0.26464) + dv3*(iteration * .01)
        dc4c = (1.0 / 0.06184) + dv4*(iteration * .01)
        dc5c = (1.0 / 0.07423) + dv5*(iteration * .01)
        dctotc = dc1c+dc2c+dc3c+dc4c+dc5c
        dc1c = dc1c/dctotc
        dc2c = dc2c/dctotc
        dc3c = dc3c/dctotc
        dc4c = dc4c/dctotc
        dc5c = dc5c/dctotc

        dc1d = (1.0 / 0.32169) + dv1*(iteration * .01)
        dc2d = (1.0 / 0.27721) + dv2*(iteration * .01)
        dc3d = (1.0 / 0.26488) + dv3*(iteration * .01)
        dc4d = (1.0 / 0.06218) + dv4*(iteration * .01)
        dc5d = (1.0 / 0.07404) + dv5*(iteration * .01)
        dctotd = dc1d+dc2d+dc3d+dc4d+dc5d
        dc1d = dc1d/dctotd
        dc2d = dc2d/dctotd
        dc3d = dc3d/dctotd
        dc4d = dc4d/dctotd
        dc5d = dc5d/dctotd

        dc1e = (1.0 / 0.32217) + dv1*(iteration * .01)
        dc2e = (1.0 / 0.27685) + dv2*(iteration * .01)
        dc3e = (1.0 / 0.26437) + dv3*(iteration * .01)
        dc4e = (1.0 / 0.06264) + dv4*(iteration * .01)
        dc5e = (1.0 / 0.07396) + dv5*(iteration * .01)
        dctote = dc1e+dc2e+dc3e+dc4e+dc5e
        dc1e = dc1e/dctote
        dc2e = dc2e/dctote
        dc3e = dc3e/dctote
        dc4e = dc4e/dctote
        dc5e = dc5e/dctote

c UPPER BOUND VALUES FOR VARIABLE n (1-5) AND
c YEARS 1-5 (a,b,c,d,e)

        ub1a = 35438.0
        ub2a = 31864.0
        ub3a = 32787.0
        ub4a = 6895.0
        ub5a = 8643.0
        ub1b = 35200.0
        ub2b = 30531.0
        ub3b = 31032.0
        ub4b = 6908.0
        ub5b = 8390.0
        ub1c = 34759.0
        ub2c = 30136.0
        ub3c = 28657.0
        ub4c = 6697.0
        ub5c = 8038.0
        ub1d = 34601.0
        ub2d = 29817.0
        ub3d = 28491.0
        ub4d = 6688.0
        ub5d = 7964.0
        ub1e = 34403.0
        ub2e = 29563.0
        ub3e = 28231.0
        ub4e = 6689.0
        ub5e = 7898.0

c DUMMY RHS VALUES FOR YEARS 1-5 (a-e)
        drhsa = 115627 - (1*cutsize)
        drhsb = 112061 - (2*cutsize)
        drhsc = 108287 - (3*cutsize)
        drhsd = 107561 - (4*cutsize)
```

63

```
        drhse = 106784 - (5*cutsize)

c ARRAYS USED TO CHANGE THE BOUNDS
        ind(1)  = 0                              .
        ind(2)  = 0
        ind(3)  = 1
        ind(4)  = 1
        ind(5)  = 2
        ind(6)  = 2
        ind(7)  = 3
        ind(8)  = 3
        ind(9)  = 4
        ind(10) = 4
        lu(1)   = 'L'
        lu(2)   = 'U'
        lu(3)   = 'L'
        lu(4)   = 'U'
        lu(5)   = 'L'
        lu(6)   = 'U'
        lu(7)   = 'L'
        lu(8)   = 'U'
        lu(9)   = 'L'
        lu(10)  = 'U'


c OPEN A FILE TO SAVE THE OUTPUT TO
        open(1,file='RUN26.out',status='unknown')

C**************** ORIGINAL PROBLEM ********************************

c Part I - Loading, optimizing and obtaining a solution to
c original problem shown in Section 4.0 of the document
c "Using the Callable Library."

c NUL character appended to strings as required by C

        status = iloadp ('example'//char(0),
        . mac, mar, 0, objsen, objx, rhsx,
        . senx, matbeg, matcnt, matind, matval,
        . bdl, bdu, idummy, idummy, idummy, idummy, idummy,
        . idummy, ddummy, datanm//char(0), objnm//char(0),
        . rhsnm//char(0), rngnm//char(0), bndnm//char(0),
        . cstore, rstore, estore, macsz, marsz, matsz,
        . 0, 0, cstsz, rstsz, 0, namlen)
        if (status .ne. 0) goto 99000

c CHANGE THE COEFFICIENTS TO INCLUDE THE CHANGE IN PARAMETERS

        dub1a = ub1a
        dub2a = ub2a
        dub3a = ub3a
        dub4a = ub4a
        dub5a = ub5a
        dlb1a = dnint(prcnt * dub1a)
        dlb2a = dnint(prcnt * dub2a)
        dlb3a = dnint(prcnt * dub3a)        .
        dlb4a = dnint(prcnt * dub4a)
        dlb5a = dnint(prcnt * dub5a)

        bd(1)   = dlb1a
        bd(2)   = dub1a
        bd(3)   = dlb2a
        bd(4)   = dub2a
        bd(5)   = dlb3a
        bd(6)   = dub3a
        bd(7)   = dlb4a
        bd(8)   = dub4a
        bd(9)   = dlb5a
        bd(10)  = dub5a
```

64

```fortran
      status = iccoef(-1,0,dc1a)
      if(status.ne.0) goto 99000
      status = iccoef(-1,1,dc2a)
      if(status.ne.0) goto 99000
      status = iccoef(-1,2,dc3a)
      if(status.ne.0) goto 99000
      status = iccoef(-1,3,dc4a)
      if(status.ne.0) goto 99000
      status = iccoef(-1,4,dc5a)
      if(status.ne.0) goto 99000

      status = iccoef(0,-1,drhsa)
      if(status.ne.0) goto 99000

      status = icbds(cnt,ind,lu,bd)
      if(status.ne.0) goto 99000

c  SET ITERATION LOGGING TO LOG EVERY FACTORIZATION
      status = sitfoi (1, toosmall, toobig)
      if (status .ne. 0) goto 99000

c  SET ITERATION LIMIT TO 10000 OR LARGEST VALUE POSSIBLE
      status = sitlim (10000, toosmall, toobig)
      if (status .ne. 0) status = sitlim (toobig, toosmall, toobig)

c WRITE THE LP TO FILE
c     status=ilpwr('RUN26a.lp'//char(0))
c     if (status .ne. 0) goto 99000

c  OPTIMIZE THE FIRST PROBLEM AND OBTAIN THE SOLUTION
      status = iopt ()
      if (status .ne. 0) goto 99000

      status = isolut (lpstat, obj, x, pi, slack, dj,
     .           1, 1, 1, 1, 1 )
      if (status .ne. 0) goto 99000
      j = j + 1
      write(*,*) j

c  WRITE THE FINAL VALUE AND DECISION VARIABLE VALUES
      write (1, 92) iteration, j, obj, x(1), x(2), x(3), x(4), x(5)

c***************** SECOND OPTIMIZATION *****************************

c CHANGE THE COEFFICIENTS, CONSTRAINTS, AND BOUNDS

      totcut1 = totcut1 + (dub1a - x(1))
      totcut2 = totcut2 + (dub2a - x(2))
      totcut3 = totcut3 + (dub3a - x(3))
      totcut4 = totcut4 + (dub4a - x(4))
      totcut5 = totcut5 + (dub5a - x(5))

      dub1b = ub1b - totcut1
      dub2b = ub2b - totcut2
      dub3b = ub3b - totcut3
      dub4b = ub4b - totcut4
      dub5b = ub5b - totcut5
      dlb1b = dnint(prcnt * dub1b)
      dlb2b = dnint(prcnt * dub2b)
      dlb3b = dnint(prcnt * dub3b)
      dlb4b = dnint(prcnt * dub4b)
      dlb5b = dnint(prcnt * dub5b)

      bd(1)  = dlb1b
      bd(2)  = dub1b
      bd(3)  = dlb2b
      bd(4)  = dub2b
      bd(5)  = dlb3b
```

65

```
      bd(6)   = dub3b
      bd(7)   = dlb4b
      bd(8)   = dub4b
      bd(9)   = dlb5b
      bd(10)  = dub5b

      status = iccõef(-1,0,dc1b)
      if(status.ne.0) goto 99000
      status = iccoef(-1,1,dc2b)
      if(status.ne.0) goto 99000
      status = iccoef(-1,2,dc3b)
      if(status.ne.0) goto 99000
      status = iccoef(-1,3,dc4b)
      if(status.ne.0) goto 99000
      status = iccoef(-1,4,dc5b)
      if(status.ne.0) goto 99000

      status = iccoef(0,-1,drhsb)
      if(status.ne.0) goto 99000

      status = icbds(cnt,ind,lu,bd)
      if(status.ne.0) goto 99000

c  SET ITERATION LOGGING TO LOG EVERY FACTORIZATION
      status = sitfoi (1, toosmall, toobig)
      if (status .ne. 0) goto 99000

c  SET ITERATION LIMIT TO 10000 OR LARGEST VALUE POSSIBLE
      status = sitlim (10000, toosmall, toobig)
      if (status .ne. 0) status = sitlim (toobig, toosmall, toobig)

c WRITE THE LP TO FILE
c     status=ilpwr('RUN26b.lp'//char(0))
c     if (status .ne. 0) goto 99000

c  OPTIMIZE THE SECOND PROBLEM AND OBTAIN THE SOLUTION
      status = iopt ()
      if (status .ne. 0) goto 99000

      status = isolut (lpstat, obj, x, pi, slack, dj,
     .            1, 1, 1, 1, 1 )
      if (status .ne. 0) goto 99000
      j = j + 1
      write(*,*) j

c  WRITE THE FINAL VALUE AND DECISION VARIABLE VALUES
      write (1, 92) iteration, j, obj, x(1), x(2), x(3), x(4), x(5)

c***************** THIRD OPTIMIZATION ****************************

c CHANGE THE COEFFICIENTS, CONSTRAINTS, AND BOUNDS

      totcut1 = totcut1 + (dub1b - x(1))
      totcut2 = totcut2 + (dub2b - x(2))
      totcut3 = totcut3 + (dub3b - x(3))
      totcut4 = totcut4 + (dub4b - x(4))
      totcut5 = totcut5 + (dub5b - x(5))

      dub1c = ub1c - totcut1
      dub2c = ub2c - totcut2
      dub3c = ub3c - totcut3
      dub4c = ub4c - totcut4
      dub5c = ub5c - totcut5
      dlb1c = dnint(prcnt * dub1c)
      dlb2c = dnint(prcnt * dub2c)
      dlb3c = dnint(prcnt * dub3c)
      dlb4c = dnint(prcnt * dub4c)
      dlb5c = dnint(prcnt * dub5c)
```

66

```
      bd(1)   = dlb1c
      bd(2)   = dub1c
      bd(3)   = dlb2c
      bd(4)   = dub2c
      bd(5)   = dlb3c
      bd(6)   = dub3c
      bd(7)   = dlb4c
      bd(8)   = dub4c
      bd(9)   = dlb5c
      bd(10)  = dub5c

      status = iccoef(-1,0,dc1c)
      if(status.ne.0) goto 99000
      status = iccoef(-1,1,dc2c)
      if(status.ne.0) goto 99000
      status = iccoef(-1,2,dc3c)
      if(status.ne.0) goto 99000
      status = iccoef(-1,3,dc4c)
      if(status.ne.0) goto 99000
      status = iccoef(-1,4,dc5c)
      if(status.ne.0) goto 99000

      status = iccoef(0,-1,drhsc)
      if(status.ne.0) goto 99000

      status = icbds(cnt,ind,lu,bd)
      if(status.ne.0) goto 99000

c  SET ITERATION LOGGING TO LOG EVERY FACTORIZATION
      status = sitfoi (1, toosmall, toobig)
      if (status .ne. 0) goto 99000

c  SET ITERATION LIMIT TO 10000 OR LARGEST VALUE POSSIBLE
      status = sitlim (10000, toosmall, toobig)
      if (status .ne. 0) status = sitlim (toobig, toosmall, toobig)

c WRITE THE LP TO FILE
c      status=ilpwr('RUN26c.lp'//char(0))
c      if (status .ne. 0) goto 99000

c OPTIMIZE THE THIRD PROBLEM AND OBTAIN THE SOLUTION
      status = iopt ()
      if (status .ne. 0) goto 99000

      status = isolut (lpstat, obj, x, pi, slack, dj,
     .           1, 1, 1, 1, 1 )
      if (status .ne. 0) goto 99000
      j = j + 1
      write(*,*) j

c  WRITE THE FINAL VALUE AND DECISION VARIABLE VALUES
      write (1, 92) iteration, j, obj, x(1), x(2), x(3), x(4), x(5)

c****************** FOURTH OPTIMIZATION ****************************
c CHANGE THE COEFFICIENTS, CONSTRAINTS, AND BOUNDS

      totcut1 = totcut1 + (dub1c - x(1))
      totcut2 = totcut2 + (dub2c - x(2))
      totcut3 = totcut3 + (dub3c - x(3))
      totcut4 = totcut4 + (dub4c - x(4))
      totcut5 = totcut5 + (dub5c - x(5))

      dub1d = ub1d - totcut1
      dub2d = ub2d - totcut2
      dub3d = ub3d - totcut3
      dub4d = ub4d - totcut4
      dub5d = ub5d - totcut5
      dlb1d = dnint(prcnt * dub1d)
      dlb2d = dnint(prcnt * dub2d)
```

67

```
      dlb3d = dnint(prcnt * dub3d)
      dlb4d = dnint(prcnt * dub4d)
      dlb5d = dnint(prcnt * dub5d)

      bd(1)   = dlb1d
      bd(2)   = dub1d
      bd(3)   = dlb2d
      bd(4)   = dub2d
      bd(5)   = dlb3d
      bd(6)   = dub3d
      bd(7)   = dlb4d
      bd(8)   = dub4d
      bd(9)   = dlb5d
      bd(10)  = dub5d

      status = iccoef(-1,0,dc1d)
      if(status.ne.0) goto 99000
      status = iccoef(-1,1,dc2d)
      if(status.ne.0) goto 99000
      status = iccoef(-1,2,dc3d)
      if(status.ne.0) goto 99000
      status = iccoef(-1,3,dc4d)
      if(status.ne.0) goto 99000
      status = iccoef(-1,4,dc5d)
      if(status.ne.0) goto 99000

      status = iccoef(0,-1,drhsd)
      if(status.ne.0) goto 99000

      status = icbds(cnt,ind,lu,bd)
      if(status.ne.0) goto 99000

c  SET ITERATION LOGGING TO LOG EVERY FACTORIZATION
      status = sitfoi (1, toosmall, toobig)
      if (status .ne. 0) goto 99000

c  SET ITERATION LIMIT TO 10000 OR LARGEST VALUE POSSIBLE
      status = sitlim (10000, toosmall, toobig)
      if (status .ne. 0) status = sitlim (toobig, toosmall, toobig)

c WRITE THE LP TO FILE
c     status=ilpwr('RUN26d.lp'//char(0))
c     if (status .ne. 0) goto 99000

c  OPTIMIZE THE FOURTH PROBLEM AND OBTAIN THE SOLUTION
      status = iopt ()
      if (status .ne. 0) goto 99000

      status = isolut (lpstat, obj, x, pi, slack, dj,
     .           1, 1, 1, 1, 1 )
      if (status .ne. 0) goto 99000
      j = j + 1
      write(*,*) j

c  WRITE THE FINAL VALUE AND DECISION VARIABLE VALUES
      write (1, 92) iteration, j, obj, x(1), x(2), x(3), x(4), x(5)

c***************** FIFTH OPTIMIZATION ****************************
c CHANGE THE COEFFICIENTS, CONSTRAINTS, AND BOUNDS

      totcut1 = totcut1 + (dub1d - x(1))
      totcut2 = totcut2 + (dub2d - x(2))
      totcut3 = totcut3 + (dub3d - x(3))
      totcut4 = totcut4 + (dub4d - x(4))
      totcut5 = totcut5 + (dub5d - x(5))

      dub1e = ub1e - totcut1
      dub2e = ub2e - totcut2
      dub3e = ub3e - totcut3
```

68

```
       dub4e = ub4e - totcut4
       dub5e = ub5e - totcut5
       dlb1e = dnint(prcnt * dub1e)
       dlb2e = dnint(prcnt * dub2e)
       dlb3e = dnint(prcnt* dub3e)
       dlb4e = dnint(prcnt * dub4e)
       dlb5e = dnint(prcnt * dub5e)

       bd(1)   = dlb1e
       bd(2)   = dub1e
       bd(3)   = dlb2e
       bd(4)   = dub2e
       bd(5)   = dlb3e
       bd(6)   = dub3e
       bd(7)   = dlb4e
       bd(8)   = dub4e
       bd(9)   = dlb5e
       bd(10)  = dub5e

       status = iccoef(-1,0,dc1e)
       if(status.ne.0) goto 99000
       status = iccoef(-1,1,dc2e)
       if(status.ne.0) goto 99000
       status = iccoef(-1,2,dc3e)
       if(status.ne.0) goto 99000
       status = iccoef(-1,3,dc4e)
       if(status.ne.0) goto 99000
       status = iccoef(-1,4,dc5e)
       if(status.ne.0) goto 99000

       status = iccoef(0,-1,drhse)
       if(status.ne.0) goto 99000

       status = icbds(cnt,ind,lu,bd)
       if(status.ne.0) goto 99000

c  SET ITERATION LOGGING TO LOG EVERY FACTORIZATION
       status = sitfoi (1, toosmall, toobig)
       if (status .ne. 0) goto 99000

c  SET ITERATION LIMIT TO 10000 OR LARGEST VALUE POSSIBLE
       status = sitlim (10000, toosmall, toobig)
       if (status .ne. 0) status = sitlim (toobig, toosmall, toobig)

c  WRITE THE LP TO FILE
c      status=ilpwr('RUN26e.lp'//char(0))
c      if (status .ne. 0) goto 99000

c  OPTIMIZE THE FIFTH PROBLEM AND OBTAIN THE SOLUTION
       status = iopt ()
       if (status .ne. 0) goto 99000

       status = isolut (lpstat, obj, x, pi, slack, dj,
       .            1, 1, 1, 1, 1 )
       if (status .ne. 0) goto 99000
       j = j + 1
       write(*,*) j

c  WRITE THE FINAL VALUE AND DECISION VARIABLE VALUES
       write (1, 92) iteration, j, obj, x(1), x(2), x(3), x(4), x(5)
       write (1,*)
c******************** FINISH SUBROUTINE ***************************

c  CLOSE THE CPLEX LOG FILE.
       status = slogfc ()

c  FREE CPLEX MEMORY
       status = iunlod ()
```

69

```
      x(1) = 0
      x(2) = 0
      x(3) = 0
      x(4) = 0
      x(5) = 0

      return

99000 continue
      write (*, *) 'Error, status = ', status

   92 format (i3,i3,f18.6,f10.2,f10.2,f10.2,f10.2,f10.2)

      end
```

Appendix B. Relative Weights Assigned in Objective Function at Points of Interest

The following results indicate an MEB's average objective function coefficient value over the five optimizations performed within one iteration. Only the points of interest are presented, along with the initial average at iteration 0. Since the objective function coefficients were normalized, these average values represent the percentage of total weight that a single MEB was assigned during that iteration. Each table is labeled above by its basecase index, direction vector applied, size of the steps taken, and level of lower bound constraints. Runs that did not have changes in the manpower allocations were not included in this appendix.

Initial State: 1   Step Size: 0.01
Direction Vector: 2   Lower Constraint % : 97

ITERATION:	0	3	4	6
BOS	0.3171	0.2831	0.2734	0.2557
PM	0.2761	0.2733	0.2725	0.271
S&IO	0.2709	0.2687	0.268	0.2668
S&T	0.0616	0.0818	0.0876	0.098
T& E	0.0743	0.0931	0.0986	0.1083

Initial State: 1   Step Size: 0.01
Direction Vector: 3   Lower Constraint % : 97

ITERATION:	0	1	4	5	24	25	26
BOS	0.317	0.308	0.283	0.276	0.184	0.181	0.178
PM	0.276	0.278	0.282	0.284	0.300	0.301	0.301
S&IO	0.271	0.263	0.242	0.236	0.158	0.155	0.152
S&T	0.062	0.070	0.091	0.097	0.175	0.178	0.181
T& E	0.074	0.082	0.102	0.108	0.183	0.185	0.188

ITERATION:	0	1	2	3	24	25	26
BOS	0.317	0.305	0.294	0.283	0.162	0.159	0.155
PM	0.276	0.285	0.293	0.300	0.386	0.388	0.390
S&IO	0.271	0.260	0.251	0.242	0.138	0.135	0.133
S&T	0.062	0.069	0.076	0.082	0.154	0.156	0.158
T& E	0.074	0.081	0.087	0.093	0.160	0.162	0.164

ITERATION:	0	2	3	4	6
BOS	0.317	0.311	0.308	0.305	0.299
PM	0.276	0.271	0.268	0.265	0.260
S&IO	0.271	0.285	0.292	0.299	0.312
S&T	0.062	0.060	0.060	0.059	0.058
T& E	0.074	0.073	0.072	0.071	0.070

ITERATION:	0	1
BOS	0.317	0.319
PM	0.276	0.249
S&IO	0.271	0.270
S&T	0.062	0.080
T& E	0.074	0.082

ITERATION:	0	3	5	7
BOS	0.317	0.444	0.495	0.532
PM	0.276	0.220	0.198	0.181
S&IO	0.271	0.195	0.164	0.142
S&T	0.062	0.066	0.068	0.069
T& E	0.074	0.075	0.075	0.076

Initial State: 1 Step Size: 0.01

Direction Vector: 9 Lower Constraint % : 97

ITERATION:	0	1
BOS	0.317	0.366
PM	0.276	0.260
S&IO	0.271	0.238
S&T	0.062	0.063
T& E	0.0743	0.074

Initial State: 1 Step Size: 0.01

Direction Vector: 0 Lower Constraint % : 97

ITERATION:	0	2
BOS	0.317	0.424
PM	0.276	0.226
S&IO	0.271	0.238
S&T	0.062	0.050
T& E	0.074	0.061

Initial State: 1 Step Size: 0.01

Direction Vector: 11 Lower Constraint % : 97

ITERATION:	0	4	5	7	8	9	10	11
BOS	0.317	0.253	0.242	0.225	0.217	0.211	0.205	0.199
PM	0.276	0.241	0.235	0.225	0.221	0.217	0.214	0.211
S&IO	0.271	0.206	0.195	0.177	0.169	0.162	0.156	0.151
S&T	0.062	0.162	0.178	0.206	0.217	0.228	0.237	0.245
T& E	0.074	0.139	0.150	0.168	0.175	0.182	0.188	0.194

Initial State: 1 Step Size: 0.01

Direction Vector: 2 Lower Constraint % : 95

ITERATION:	0	4	5	6	24	25	26
BOS	0.3171	0.2734	0.264	0.256	0.162	0.159	0.155
PM	0.2761	0.2725	0.272	0.271	0.263	0.263	0.263
S&IO	0.2709	0.268	0.267	0.267	0.261	0.260	0.260
S&T	0.0616	0.0876	0.093	0.098	0.154	0.156	0.158
T& E	0.0743	0.0986	0.104	0.108	0.160	0.162	0.164

Initial State: 1 Step Size: 0.01
Direction Vector: 3 Lower Constraint % : 95

ITERATION:	0	1	5	21	25	26
BOS	0.317	0.308	0.276	0.195	0.181	0.178
PM	0.276	0.278	0.284	0.298	0.301	0.301
S&IO	0.271	0.263	0.236	0.166	0.155	0.152
S&T	0.062	0.070	0.097	0.167	0.178	0.181
T& E	0.074	0.082	0.108	0.174	0.185	0.188

Initial State: 1 Step Size: 0.01
Direction Vector: 4 Lower Constraint % : 95

ITERATION:	0	1	3	21	25	26
BOS	0.3171	0.305	0.283	0.172	0.159	0.155
PM	0.2761	0.285	0.300	0.378	0.388	0.390
S&IO	0.2709	0.260	0.242	0.147	0.135	0.133
S&T	0.0616	0.069	0.082	0.148	0.156	0.158
T& E	0.0743	0.081	0.093	0.155	0.162	0.164

Initial State: 1 Step Size: 0.01
Direction Vector: 5 Lower Constraint % : 95

ITERATION:	0	2	6
BOS	0.3171	0.3109	0.2992
PM	0.2761	0.271	0.260
S&IO	0.2709	0.285	0.312
S&T	0.0616	0.060	0.058
T& E	0.0743	0.073	0.070

Initial State: 1 Step Size: 0.01
Direction Vector: 6 Lower Constraint % : 95

ITERATION:	0	1	11	21
BOS	0.3171	0.319	0.327	0.329
PM	0.2761	0.249	0.146	0.117
S&IO	0.2709	0.270	0.268	0.268
S&T	0.0616	0.080	0.148	0.167
T& E	0.0743	0.082	0.111	0.119

74

ITERATION:	0	1	21
BOS	0.3171	0.3691	0.648
PM	0.2761	0.253	0.130
S&IO	0.2709	0.240	0.073
S&T	0.0616	0.063	0.073
T& E	0.0743	0.075	0.076

Initial State: 1   Step Size: 0.01

Direction Vector: 9   Lower Constraint % : 95

ITERATION:	0	1	21
BOS	0.317	0.366	0.613
PM	0.276	0.260	0.177
S&IO	0.271	0.238	0.069
S&T	0.062	0.063	0.069
T& E	0.074	0.074	0.072

Initial State: 1   Step Size: 0.01

Direction Vector: 0   Lower Constraint % : 95

ITERATION:	0	2
BOS	0.3171	0.4238
PM	0.2761	0.2263
S&IO	0.2709	0.238
S&T	0.0616	0.050
T& E	0.0743	0.061

Initial State: 1   Step Size: 0.01

Direction Vector: 11   Lower Constraint % : 95

ITERATION:	0	1	5	6	7	8	9
BOS	0.317	0.297	0.242	0.233	0.225	0.217	0.211
PM	0.276	0.265	0.235	0.230	0.225	0.221	0.217
S&IO	0.271	0.250	0.195	0.185	0.177	0.169	0.162
S&T	0.062	0.093	0.178	0.193	0.206	0.217	0.228
T& E	0.074	0.095	0.150	0.160	0.168	0.175	0.182

Initial State: 1   Step Size: 0.1
Direction Vector: 2   Lower Constraint % : 97

ITERATION:	0	1
BOS	0.3171	0.2265
PM	0.2761	0.2686
S&IO	0.2709	0.2649
S&T	0.0616	0.1154
T& E	0.0743	0.1245

Initial State: 1   Step Size: 0.1
Direction Vector: 3   Lower Constraint % : 97

ITERATION:	0	1	3
BOS	0.3171	0.244	0.1669
PM	0.2761	0.2893	0.3032
S&IO	0.2709	0.2084	0.1426
S&T	0.0616	0.1243	0.190
T& E	0.0743	0.1341	0.197

Initial State: 1   Step Size: 0.1
Direction Vector: 4   Lower Constraint % : 97

ITERATION:	0	1	3
BOS	0.3171	0.227	0.144
PM	0.2761	0.340	0.398
S&IO	0.2709	0.194	0.123
S&T	0.0616	0.115	0.164
T& E	0.0743	0.125	0.170

Initial State: 1   Step Size: 0.1
Direction Vector: 5   Lower Constraint % : 97

ITERATION:	0	1
BOS	0.3171	0.2883
PM	0.2761	0.251
S&IO	0.2709	0.3371
S&T	0.0616	0.056
T& E	0.0743	0.0676

76

Initial State: 1   Step Size: 0.1
Direction Vector: 6   Lower Constraint % : 97

ITERATION:	0	1
BOS	0.3171	0.327
PM	0.2761	0.150
S&IO	0.2709	0.268
S&T	0.0616	0.145
T& E	0.0743	0.110

Initial State: 1   Step Size: 0.1
Direction Vector: 8   Lower Constraint % : 97

ITERATION:	0	1
BOS	0.317	0.573
PM	0.276	0.164
S&IO	0.271	0.118
S&T	0.062	0.070
T& E	0.0743	0.0758

Initial State: 1   Step Size: 0.1
Direction Vector: 9   Lower Constraint % : 97

ITERATION:	0	1
BOS	0.3171	0.5487
PM	0.2761	0.1984
S&IO	0.2709	0.1129
S&T	0.0616	0.0673
T& E	0.0743	0.0726

Initial State: 1   Step Size: 0.1
Direction Vector: 0   Lower Constraint % : 97

ITERATION:	0	1
BOS	0.3171	0.6271
PM	0.2761	0.1315
S&IO	0.2709	0.1766
S&T	0.0616	0.0293
T& E	0.0743	0.0354

Initial State: 1   Step Size: 0.1

Direction Vector: 11   Lower Constraint % : 97

ITERATION:	0	1	2
BOS	0.317	0.205	0.167
PM	0.276	0.214	0.193
S&IO	0.271	0.156	0.118
S&T	0.062	0.237	0.296
T& E	0.074	0.188	0.226

Initial State: 1   Step Size: 0.1

Direction Vector: 2   Lower Constraint % : 95

ITERATION:	0	1	3
BOS	0.317	0.227	0.144
PM	0.276	0.269	0.262
S&IO	0.271	0.265	0.259
S&T	0.062	0.115	0.164
T& E	0.074	0.125	0.170

Initial State: 1   Step Size: 0.1

Direction Vector: 3   Lower Constraint % : 95

ITERATION:	0	1	3
BOS	0.317	0.244	0.167
PM	0.276	0.289	0.303
S&IO	0.271	0.208	0.143
S&T	0.062	0.124	0.190
T& E	0.074	0.134	0.197

Initial State: 1   Step Size: 0.1

Direction Vector: 4   Lower Constraint % : 95

ITERATION:	0	1	3
BOS	0.317	0.227	0.144
PM	0.276	0.340	0.398
S&IO	0.271	0.194	0.123
S&T	0.062	0.115	0.164
T& E	0.074	0.125	0.170

Initial State: 1   Step Size: 0.1
Direction Vector: 5   Lower Constraint % : 95

ITERATION:	0	1
BOS	0.3171	0.2883
PM	0.2761	0.251
S&IO	0.2709	0.3371
S&T	0.0616	0.056
T& E	0.0743	0.0676

Initial State: 1   Step Size: 0.1
Direction Vector: 6   Lower Constraint % : 95

ITERATION:	0	1	2	3
BOS	0.317	0.327	0.329	0.330
PM	0.276	0.150	0.119	0.105
S&IO	0.271	0.268	0.268	0.267
S&T	0.062	0.145	0.165	0.175
T& E	0.074	0.110	0.119	0.123

Initial State: 1   Step Size: 0.1
Direction Vector: 8   Lower Constraint % : 95

ITERATION:	0	1	3
BOS	0.317	0.573	0.677
PM	0.276	0.164	0.118
S&IO	0.271	0.118	0.055
S&T	0.062	0.070	0.074
T& E	0.074	0.076	0.076

Initial State: 1   Step Size: 0.1
Direction Vector: 9   Lower Constraint % : 95

ITERATION:	0	1	3
BOS	0.317	0.549	0.638
PM	0.276	0.198	0.168
S&IO	0.271	0.113	0.052
S&T	0.062	0.067	0.070
T& E	0.074	0.073	0.072

79

| Initial State: 1 | | Step Size: 0.1 |
| Direction Vector: 0 | | Lower Constraint % : 95 |

ITERATION:	0	1
BOS	0.3171	0.6271
PM	0.2761	0.1315
S&IO	0.2709	0.1766
S&T	0.0616	0.0293
T& E	0.0743	0.0354

| Initial State: 1 | | Step Size: 0.1 |
| Direction Vector: 11 | | Lower Constraint % : 95 |

ITERATION:	0	1	2
BOS	0.3171	0.2046	0.1668
PM	0.2761	0.2141	0.1933
S&IO	0.2709	0.1563	0.1178
S&T	0.0616	0.2369	0.2958
T& E	0.0743	0.1881	0.2263

| Initial State: | 2 | Step Size: | 0.003 |
| Direction Vector: | 4 | Lower Constraint % : | 97 |

ITERATION:	0	81
BOS	0.079	0.040
PM	0.090	0.292
S&IO	0.092	0.047
S&T	0.404	0.328
T& E	0.335	0.293

| Initial State: | 2 | Step Size: | 0.003 |
| Direction Vector: | 5 | Lower Constraint % : | 97 |

ITERATION:	0	1	82
BOS	0.079	0.078	0.063
PM	0.090	0.095	0.072
S&IO	0.092	0.091	0.271
S&T	0.404	0.402	0.325
T& E	0.335	0.334	0.269

Initial State:	2			Step Size:	0.003	
Direction Vector:	6			Lower Constraint % :	97	

ITERATION:	0	1	3	28	41	56
BOS	0.079	0.090	0.109	0.221	0.244	0.261
PM	0.090	0.089	0.087	0.077	0.075	0.073
S&IO	0.092	0.100	0.113	0.189	0.205	0.217
S&T	0.404	0.396	0.380	0.290	0.272	0.258
T& E	0.335	0.326	0.311	0.223	0.204	0.191

Initial State:	2			Step Size:	0.003
Direction Vector:	7			Lower Constraint % :	97

ITERATION:	0	1	10	13
BOS	0.079	0.104	0.267	0.303
PM	0.090	0.089	0.085	0.084
S&IO	0.092	0.091	0.086	0.085
S&T	0.404	0.391	0.306	0.287
T& E	0.335	0.324	0.257	0.242

Initial State:	2			Step Size:	0.003
Direction Vector:	8			Lower Constraint % :	97

ITERATION:	0	1	10	13
BOS	0.079	0.104	0.272	0.311
PM	0.090	0.090	0.086	0.086
S&IO	0.092	0.089	0.066	0.061
S&T	0.404	0.392	0.312	0.294
T& E	0.335	0.325	0.263	0.248

Initial State:	2			Step Size:	0.003
Direction Vector:	9			Lower Constraint % :	97

ITERATION:	0	1	10	13
BOS	0.079	0.104	0.267	0.303
PM	0.090	0.092	0.106	0.109
S&IO	0.092	0.088	0.065	0.060
S&T	0.404	0.391	0.306	0.287
T& E	0.335	0.324	0.257	0.242

| Initial State: | 2 | | | Step Size: | 0.003 | | |
| Direction Vector: | 0 | | | Lower Constraint % : | 97 | | |

ITERATION:	0	1	4	9	12	82	87
BOS	0.079	0.105	0.175	0.269	0.314	0.685	0.695
PM	0.090	0.087	0.080	0.070	0.065	0.024	0.023
S&IO	0.092	0.092	0.092	0.092	0.092	0.091	0.091
S&T	0.404	0.391	0.357	0.312	0.290	0.109	0.104
T& E	0.335	0.324	0.296	0.258	0.240	0.090	0.087

| Initial State: | 2 | | Step Size: | 0.003 |
| Direction Vector: | 5 | | Lower Constraint % : | 95 |

ITERATION:	0	1
BOS	0.079	0.078
PM	0.090	0.090
S&IO	0.092	0.095
S&T	0.404	0.403
T& E	0.335	0.334

| Initial State: | 2 | | Step Size: | 0.003 |
| Direction Vector: | 6 | | Lower Constraint % : | 95 |

ITERATION:	0	1	3	41
BOS	0.079	0.090	0.109	0.244
PM	0.090	0.089	0.087	0.075
S&IO	0.092	0.100	0.113	0.205
S&T	0.404	0.396	0.380	0.272
T& E	0.335	0.326	0.311	0.204

| Initial State: | 2 | | Step Size: | 0.003 |
| Direction Vector: | 7 | | Lower Constraint % : | 95 |

ITERATION:	0	1
BOS	0.079	0.104
PM	0.090	0.089
S&IO	0.092	0.091
S&T	0.404	0.391
T& E	0.335	0.324

| Initial State: | 2 | | Step Size: | 0.003 |
| Direction Vector: | 8 | | Lower Constraint % : | 95 |

ITERATION:	0	1
BOS	0.079	0.104
PM	0.090	0.090
S&IO	0.092	0.089
S&T	0.404	0.392
T& E	0.335	0.325

| Initial State: | 2 | | Step Size: | 0.003 |
| Direction Vector: | 9 | | Lower Constraint % : | 95 |

ITERATION:	0	1	81
BOS	0.079	0.104	0.570
PM	0.090	0.092	0.131
S&IO	0.092	0.088	0.021
S&T	0.404	0.391	0.147
T& E	0.335	0.324	0.131

| Initial State: | 2 | | Step Size: | 0.003 |
| Direction Vector: | 0 | | Lower Constraint % : | 95 |

ITERATION:	0	1	82
BOS	0.079	0.105	0.685
PM	0.090	0.087	0.024
S&IO	0.092	0.092	0.091
S&T	0.404	0.391	0.109
T& E	0.335	0.324	0.090

| Initial State: | 2 | | Step Size: | 0.01 |
| Direction Vector: | 4 | | Lower Constraint % : | 97 |

ITERATION:	0	25	33
BOS	0.079	0.039	0.034
PM	0.090	0.295	0.323
S&IO	0.092	0.046	0.040
S&T	0.404	0.327	0.317
T& E	0.335	0.292	0.287

| | Initial State: | 2 | | | Step Size: | 0.01 |
| | Direction Vector: | 5 | | Lower Constraint % : | | 97 |

ITERATION:	0	1	25	33
BOS	0.079	0.078	0.063	0.059
PM	0.090	0.089	0.072	0.068
S&IO	0.092	0.101	0.274	0.317
S&T	0.404	0.400	0.323	0.304
T& E	0.335	0.332	0.268	0.252

| | Initial State: | 2 | | | | Step Size: | 0.01 |
| | Direction Vector: | 6 | | | Lower Constraint % : | | 97 |

ITERATION:	0	1	9	13	17	33
BOS	0.079	0.112	0.225	0.247	0.262	0.291
PM	0.090	0.087	0.077	0.075	0.073	0.071
S&IO	0.092	0.115	0.192	0.207	0.217	0.237
S&T	0.404	0.378	0.287	0.269	0.258	0.234
T& E	0.335	0.309	0.219	0.202	0.190	0.167

| | Initial State: | 2 | | | Step Size: | 0.01 |
| | Direction Vector: | 7 | | Lower Constraint % : | | 97 |

ITERATION:	0	1	3	4
BOS	0.079	0.157	0.267	0.307
PM	0.090	0.088	0.085	0.083
S&IO	0.092	0.089	0.086	0.085
S&T	0.404	0.363	0.306	0.285
T& E	0.335	0.303	0.257	0.240

| | Initial State: | 2 | | | Step Size: | 0.01 |
| | Direction Vector: | 8 | | Lower Constraint % : | | 97 |

ITERATION:	0	1	3	4
BOS	0.079	0.158	0.272	0.315
PM	0.090	0.089	0.086	0.086
S&IO	0.092	0.081	0.066	0.061
S&T	0.404	0.367	0.312	0.292
T& E	0.335	0.305	0.263	0.247

84

Initial State: 2   Step Size: 0.01
Direction Vector: 9   Lower Constraint % : 97

ITERATION:	0	1	3	4	25	33
BOS	0.079	0.157	0.267	0.307	0.573	0.601
PM	0.090	0.097	0.106	0.109	0.131	0.133
S&IO	0.092	0.081	0.065	0.059	0.020	0.016
S&T	0.404	0.363	0.306	0.285	0.145	0.131
T& E	0.335	0.303	0.257	0.240	0.130	0.118

Initial State: 2   Step Size: 0.01
Direction Vector: 0   Lower Constraint % : 97

ITERATION:	0	1	2	3	4	25	27	33	35
BOS	0.079	0.161	0.228	0.285	0.332	0.688	0.700	0.730	0.738
PM	0.090	0.081	0.074	0.068	0.063	0.024	0.023	0.019	0.019
S&IO	0.092	0.092	0.092	0.092	0.092	0.091	0.091	0.091	0.091
S&T	0.404	0.364	0.331	0.304	0.281	0.108	0.102	0.087	0.083
T& E	0.335	0.302	0.275	0.252	0.233	0.089	0.084	0.072	0.069

Initial State: 2   Step Size: 0.01
Direction Vector: 5   Lower Constraint % : 95

ITERATION:	0	1
BOS	0.079	0.078
PM	0.090	0.089
S&IO	0.092	0.101
S&T	0.404	0.400
T& E	0.335	0.332

Initial State: 2   Step Size: 0.01
Direction Vector: 6   Lower Constraint % : 95

ITERATION:	0	1	13	33
BOS	0.079	0.112	0.247	0.291
PM	0.090	0.087	0.075	0.071
S&IO	0.092	0.115	0.207	0.237
S&T	0.404	0.378	0.269	0.234
T& E	0.335	0.309	0.202	0.167

| Initial State: | 2 | | Step Size: | 0.01 |
| Direction Vector: | 7 | | Lower Constraint % : | 95 |

ITERATION:	0	1
BOS	0.079	0.157
PM	0.090	0.088
S&IO	0.092	0.089
S&T	0.404	0.363
T& E	0.335	0.303

| Initial State: | 2 | | Step Size: | 0.01 |
| Direction Vector: | 8 | | Lower Constraint % : | 95 |

ITERATION:	0	1
BOS	0.079	0.158
PM	0.090	0.089
S&IO	0.092	0.081
S&T	0.404	0.367
T& E	0.335	0.305

| Initial State: | 2 | | Step Size: | 0.01 |
| Direction Vector: | 9 | | Lower Constraint % : | 95 |

ITERATION:	0	1	25	33
BOS	0.079	0.157	0.573	0.601
PM	0.090	0.097	0.131	0.133
S&IO	0.092	0.081	0.020	0.016
S&T	0.404	0.363	0.145	0.131
T& E	0.335	0.303	0.130	0.118

| Initial State: | 2 | | Step Size: | 0.01 |
| Direction Vector: | 0 | | Lower Constraint % : | 95 |

ITERATION:	0	1	25	33
BOS	0.079	0.161	0.688	0.730
PM	0.090	0.081	0.024	0.019
S&IO	0.092	0.092	0.091	0.091
S&T	0.404	0.364	0.108	0.087
T& E	0.335	0.302	0.089	0.072

Bibliography

1. Bishop, Captain Steven T. <u>A Methodology to Generate Air Force Materiel Command Manpower Reduction Alternatives</u>. MS thesis, AFIT/GOR/ENS/95M-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 1995.

2. Brafford, Joe. AFMC/DR, Wright-Patterson AFB OH. Personal interview. 23 October 1995.

3. Coppins, Richard. <u>Linear Programming and Extensions</u>. New York: McGraw-Hill Book Company, 1981.

4. CPLEX. Version 3.0, IBM, 1.44MB, disk. Computer software. CPLEX Optimization, Inc., Incline Village, NV, 1989.

5. Eiger, Amy, Jonathan M. Jacobs, Donald B. Chung, and James Selsor. "The U.S. Army's Occupational Specialty Manpower Decision Support System," <u>INTERFACES</u>, 18: 57-73 (January-February 1988).

6. Microsoft Excel. Version 5.0, IBM, 1.44MB, disk. Computer software. Microsoft Corporation, Redmond, WA, 1993.

7. Gayley, James. AFMC/XPMR, Wright-Patterson AFB, OH. Personal interview. 5 December 1995.

8. Gass, Saul I. "Military Manpower Planning Models," <u>Journal of Computers and Operations Research</u>, Vol 18 No. 1: 65-73 (1991).

9. Hillier, Frederick S. and Gerald J. Lieberman. <u>Introduction to Operations Research</u>. New York: McGraw-Hill Publishing Company, 1990.

10. Karthikeyan, G. and K.N. Krishnaswamy. "Assembly Manpower Allocation Under Proportionality Constraints," <u>European Journal of Operations Research</u>, 44: 39-46 (1990).

11. Scullion, TSgt Rich. AFMC MEO/ME, Wright-Patterson AFB, OH. Personal interview. 20 October 1995.

12. Smith, Captain Sandra K. <u>A Methodology for Comparing the Value of Competing AFMC Manpower Allocation Strategies.</u> MS thesis, AFIT/GOR/ENS/95M-16. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 1995.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE March, 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE

A METHODOLOGY TO ACCESS THE AIR FORCE MATERIEL COMMAND'S EXERCISE SUPPORT PROGRAM'S WEIGHTING SCHEME

5. FUNDING NUMBERS

6. AUTHOR(S)

Gilbert D. Ness, 2nd Lieutenant, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
2750 P Street
WPAFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOR/ENS/96M-10

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

HQ AFMC/XPM (MEO/ME)
4375 Chidlaw Road
Suite C
WPAFB, OH 45433-5006

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This research performs analysis on the Air Force Materiel Command's Exercise Support Program's Weighting Scheme so that decision makers can create a more efficient and effective Resource Allocation - Integrated Process Team (RA-IPT) manpower allocation process. A linear programming (LP) model was derived from the Exercise Support Program (ESP) to assign manpower reductions throughout the Mission Element Board (MEB) organizational level in AFMC. Parametric analysis was performed on the LP model by simultaneously changing multiple objective function coefficients with the use of various direction vectors. These direction vectors were applied with the use of an incrementally increasing scalar step size and initiated from two initial states of objective function coefficient values. Throughout the analysis, multiple bounds on the LP model's constraints and different scalar step sizes were applied. Results of the analysis indicated that there were specific situations in which changes in the relative weighting scheme did effect manpower allocations to the MEBs. These results also indicated that this analysis, along with the ESP model, could allow the decision makers in the RA-IPT to become more efficient in their manpower allocation process.

14. SUBJECT TERMS

Manpower Allocation; Parametric Analysis

15. NUMBER OF PAGES
98

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

Block 1. Agency Use Only (*Leave blank*).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	-	Contract	PR	- Project
G	-	Grant	TA	- Task
PE	-	Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (*If known*)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (*NTIS only*).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.